STATISTICAL METHODS FOR COUPLING EXPERT KNOWLEDGE AND AUTOMATIC IMAGE SEGMENTATION AND REGISTRATION

A Thesis
Presented to
The Academic Faculty

by

Ivan A. Kolesov

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2013

STATISTICAL METHODS FOR COUPLING EXPERT KNOWLEDGE AND AUTOMATIC IMAGE SEGMENTATION AND REGISTRATION

Approved by:

Professor Anthony Yezzi,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Patricio Vela, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Nikil Jayant
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Allen Tannenbaum, Advisor
School of Electrical and Computer Engineering
Boston University

Professor John Xerogeanes
School of Medicine
Emory University

Date Approved: 6 December 2012

To my parents,

Alexander and Irina.

This work is not only a result of my efforts but also, in large part, of yours.

ACKNOWLEDGEMENTS

Whether knowingly or inadvertently, a number of people have supported me on the path towards earning this degree. I would like to acknowledge the people below for being in my life during these years. Thank your for making the experience enjoyable, fulfilling, and unforgettable!

- **Professor Patricio Vela:** Thank you for welcoming me into your lab and making the transition go so smoothly. Your genuine passion for research and sincere concern in the success of your students were great motivators. Your clear, systematic approach for attacking a research problem is one that I have tried to emulate and will continue to use in the future. Whenever my progress stagnated, your uncanny ability to pose relevant questions and provide suggestions propelled my work forward. I am grateful for the numerous research-related conversations as well as ones off-topic that we have had, they were both enlightening and enjoyable.

- **Professor Allen Tannenbaum:** Working with you has been a privilege. Your gentle guidance focused my research ambitions into a manageable size while giving me the freedom to pursue my research interests, and your obvious brilliance combined with a confident, personable nature always inspired a feeling of certainty in the success of each new endeavor. To me, the ultimate goal of an advisor is to mold his students into independent researchers. The variety of experiences, professional contacts, and breadth of projects I have had under your advisement have been much wider than I ever anticipated; as a result, I am confident in my ability to pose new questions and tackle open problems, thank you.

- **Professor Eldad Haber:** During the short time we worked together, I learned so much. Thank you for being patient and approachable, at a time when I was so "green."

- **My thesis committee:** Thank you to Professors Jeff Shamma, Anthony Yezzi, Nikil Jayant, and John Xerogeanes for being part of my thesis committee. Your suggestions, comments, and questions have helped me see my own work from different perspectives and clarify the content, improving this dissertation in the process.

- **My Research Team:** A strong team always achieves more than any individual, and the two of you are some of the most talented people I know. Peter, your work ethic has been a model for me and your adventurous attitude towards trying then next new idea was frequently the spark needed to escape from a "dead-end." Martin, your attention to detail and ability to efficiently locate the root of a problem are unparalleled, which combined into an approach that often ended up winning the race despite the misleading, slow appearance. Peter and Martin, I am grateful to have worked with you closely during these years, to have shared with you many a moment of distraction, whether in the form of a coffee break or a day trip out of Atlanta, and most importantly, to be able to call you my friends.

- **MINERVA and IVALab members:** It has been a pleasure to have Liang-Jia, Jehoon, Yi, Rome, Behnood, Vandana, Parya, Yifei, Gallagher, Jacob, Shawn, Xavier, Jimi, Tauseef, Eli, John, Chudy, Marin, Sarosh, Matias, Miguel, Tayo, Hassan, and Kelvin as lab-mates during my graduate career. You made the lab a truly pleasant place to be and one where ideas were shared as comfortably as plans for the upcoming weekend.

- **My friends:** Whether we were in the same city or separated by an ocean, whether we shared a laugh over dinner or a long overdue phone call, you helped me maintain

a positive outlook, and a healthy balance in my life. My friends, you know who you are, thank you!

- **My brother:** Vladimir, what can I say? You've seen this path from start to end: the good times, the stressful times, the temporary failures, and ultimately, success. Your proximity has frequently made you the victim of reading rough drafts, listening to practice presentations, and more requests for favors than I can count, and for that, thank you. See you on the other side!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Computer vision is the science of extracting contextual clues from images that allow a computer system to obtain a higher level understanding of a scene. Often, complex processes such as fire, anatomical changes/variations in human bodies, or unpredictable human behavior produce the images; in these cases, creating a model that precisely describes the process is not feasible. A common solution is to make simplifying assumptions when performing detection, segmentation, or registration tasks automatically. However, when these assumptions are not satisfied, the results are unsatisfactory. In this dissertation, we propose removing several of the typical assumptions at the cost of minimal user input and develop methods for segmentation and registration that aim to optimally make use of the user's guidance. Specifically:

1. An interactive image segmentation approach is described in which the user is coupled in a closed-loop control system with a level set segmentation algorithm. The user's expert knowledge is combined with the speed of automatic segmentation.

2. An iterative, landmark-based segmentation method is proposed to simulate an expert user's approach to segmentation of difficult structures. Here, models relating dependent anatomical structures are learned and a prediction is made for the location of the subsequent organs based on completed segmentations. After each prediction, a user can correct the segmentation with an interactive segmentation method (e.g., see Item 1).

3. A stochastic point set registration algorithm is explored. The point sets can be derived from simple user input (e.g. a threshold operation), and time consuming correspondence labeling is not required. Furthermore, common smoothness assumptions on

the non-rigid deformation field are removed.

4. An image registration technique is described that can robustly capture large image deformations. Stochastic registration is used to align two images rigidly. Then, a small number of bases capture gross, non-rigid deformations; their parameters are also optimized stochastically. Finally, small misalignments are corrected by a gradient descent optimization over densely spaced basis functions.

For future work, there are a number of promising directions. First, an interesting research direction is to develop an interactive registration technique that is able to attain ground truth results for complex images. Second, different methods for fusing label maps to form the predicted segmentation in the landmark-based approach (Item 2) should be examined. Exploring options for efficiently initializing and adaptively changing the dynamics for different registration cases is expected to improve the convergence rate of the registration techniques described. Furthermore, different constraints can be proposed to regularize the registration process.

CHAPTER I

INTRODUCTION

1.1 Computer Vision

Human perception of the environment is formed based on the traditional five senses, a classification often attributed to Aristotle: sight, smell, taste, touch, and hearing. Sight is the most important sense for humans to acquire information about their surroundings. Tasks such as recognizing objects, judging sizes and distances, and interpreting scenes are quickly and accurately performed by people everyday. However, describing the features we use and the process we go through to perform these tasks is significantly more challenging.

Computer vision is the field of study that focuses on endowing computers with the ability to analyze the real world through visual clues. Although human users can, in general, perform vision tasks faster and more accurately, computers have several superior characteristics. First, people are susceptible to *human error* that may be a result of fatigue, emotions, or distractions. Second, work hours are substantially more expensive for a person, which leads to prohibitively high cost of implementing a system (e.g., a human would not be tasked with watching a video feed of a warehouse for a potential fire). Third, it is often necessary to record an output at the completion of a vision task that is later used to perform a related action(e.g., a detection of a missing component on an assembly line leads to an insertion and soldering of this component), which is more taxing for a human than a machine.

In this dissertation, the focus is on two common image processing goals, specifically applied to medical images. The first is *segmentation*, introduced in Section 1.2, and the second is *registration*, discussed in Section 1.3. Segmentation and registration are tightly

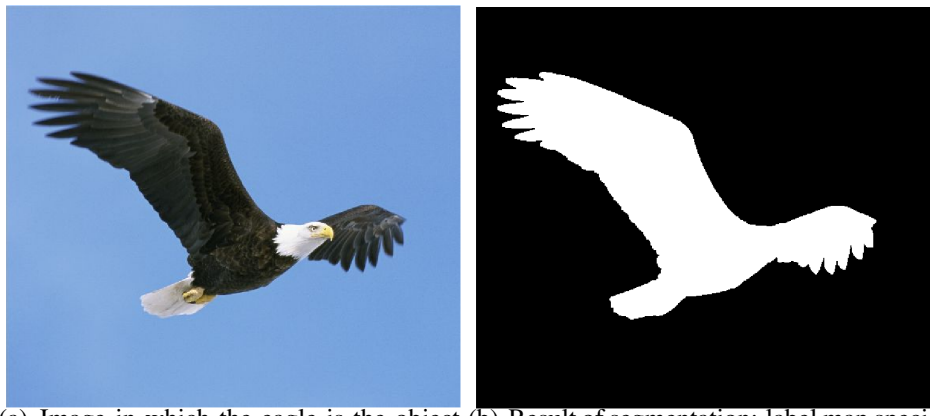
coupled because an available registration to a known image can serve as strong prior information for a segmentation approach, and a completed segmentation of a scene greatly simplifies the registration problem.

1.2 Image Segmentation

Image segmentation is the task of delineating an object of interest from the background. Segmentation is often the link between low level (e.g., feature extraction) and high level (e.g., creation of geometric models) tasks; it can be split into three categories. First are automatic methods requiring no user input, such as Otsu's method [69]. Second are semi-automatic methods that are initialized by the user; the majority methods in this category are based either on *graphs* [13] or *variational active contours*, discussed in Section 2.1. Often these methods are sensitive to initialization, providing a sufficiently close initial label is not trivial. Finally, segmentation can be interactive in which the user provides input throughout the process, guiding the algorithm in difficult image regions.

A segmentation example is shown in Fig. 1. It contains a salient object of interest with a roughly homogeneous intensity profile for the background that differs vastly from the object; in short, this is a simple case. In a real scenario, such as a medical imagery, objects often do not have intensity profiles significantly different from one another or the background, on a global scale. Surrounding clutter contributes to the regional image statistics, and a number of noise sources tend to be present. A difficult segmentation example is presented in Fig. 2; here, the structures of interest cannot be segmented based on image data alone.

One way of improving segmentation accuracy in an image such as Fig. 2 is to use a shape prior in the segmentation process. Shape models serve as prior knowledge for a segmentation algorithm. In [53], the authors built a principal component analysis (PCA) model for a structure of interest and used it to steer the segmentation by computing the likelihood of the current result. Similarly, in [79], a mean shape and shape variations are



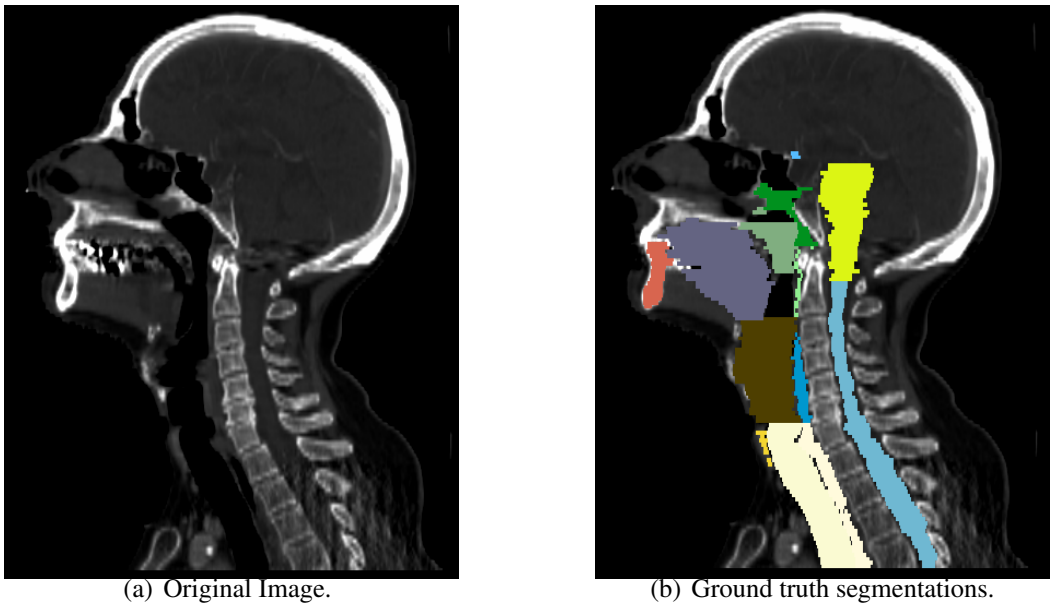
(a) Image in which the eagle is the object of interest.

(b) Result of segmentation: label map specifying which pixels belong to the foreground.

Figure 1: A figure illustrating segmentation of the image in Fig. 1(a). The output of a segmenation algorithm is a labeling for each pixel to denote if it belongs to an object of interest or the background; an example labeling is presented in Fig. 1(b).

learned, and an energy based on photometric properties of the image is regularized with with the maximum likelihood of a segmenting curve given the shape prior. The approach in [93] built on this work by creating a parametric model using PCA on the signed distance functions (SDF) and evolving the parameters to minimize an objective function. A further extension was proposed in [25], where a non-linear shape model was is made using Kernel PCA (KPCA).

Atlas based segmentation has been one of the most popular methods for medical image. This approach requires reference image(s) and the corresponding segmentation(s) to be available: it relies on accurate registration to find correspondences between an incoming image and the images in an atlas and transfers the corresponding labels according to the computed deformation field onto the query [92], [28]. If a single atlas is used, it does not take into account the variance across a population. Thus, extensions of these methods to use multiple atlases have been proposed [3], [74]. In multi-atlas segmentation, a pairwise registration of the incoming image to each of the images in the atlas is performed; this leads to a greatly increased computational complexity and raises the question of how each of the segmentations should be combined to produce a single result, which known as the label



(a) Original Image. (b) Ground truth segmentations.

Figure 2: This figure contains a image for which segmentation is considered difficult. The targets are colored differently; segmentation is complicated by the nearby objects of similar intensities and noise present in the image.

fusion problem [101], [82]. Creating shape models or atlases for segmentation involves alignment of images into a common coordinate system, which is the topic of the following section.

1.3 Registration

Registration is the task of estimating a transformation that deforms a moving data set to optimally align with a reference data set. Input data may be a raw image (i.e. a function mapping pixel locations to intensity values), or a set of points. The data can be of arbitrary dimension, but typically, two dimensional images or point sets, three dimensional volumetric images or point clouds, and four dimensional sequences of image volumes or sequences of point clouds are considered.

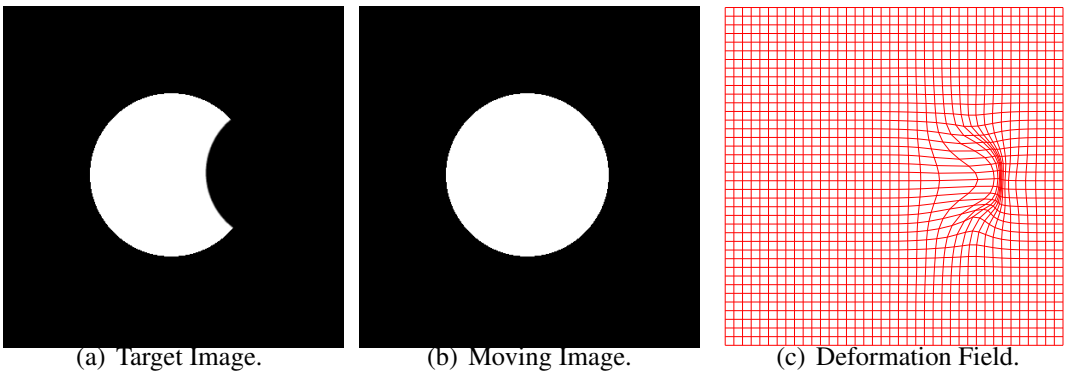
An image registration example is shown in Fig. 3: the target and moving images as well as the computed registration field are visualized. Warping the domain of the moving image according to this field results in a new image that is similar to the target. The first choice in designing a registration algorithm is selecting the similarity metric. Two simple

metrics used to directly compare images are the cross-correlation [44, 72] and the L_p norm of the difference between a moving image and a reference image [92]. These similarity metrics are not applicable for images from different sensors since intensities composing an object in one image will be different from the intensities belonging to the same object in the other image. But, a particular intensity value obtained from the first sensor is expected to consistently correlate to a different intensity value for the same object obtained by the second sensor. This premise led to the application of metrics from information theory such as joint entropy [87] and mutual information [24, 99] for image registration. To remove dependence of the registration on the area of overlap [89], normalized mutual information was proposed as a measure of similarity in [103]. Most recently, intensity distributions from two images were compared using the Bhattacharrya distance [58] for multi-modal registration.

The second feature differentiating registration techniques is the deformation model they employ. If certain assumptions can be made about the deformation relating two images, the registration problem can be simplified. This additional knowledge is useful in two ways: a field with fewer degrees of freedom is likely to be more computationally efficient, and strong regularization is imposed on the deformation, which is expected to produce fewer local minima. A rigid registration [88] is appropriate for two images of the same object, an affine transformation [41, 64] may be used as an approximation of global changes in the scene, and transformation models with higher degree of freedom (DOF), such as B-splines [81], thin plate splines (TPS) [62], or other radial basis functions [31] are applicable for complex, non-rigid, local deformations. For instance, in medical imaging, rigid registration may be sufficient for intra-subject registration and a B-spline representation required to capture local changes for inter-subject registration; choices for the similarity metric and the representation are made based on the application in hand.

The final piece for a complete registration approach is picking a method for optimizing

the similarity metric. Local minimization with gradient descent is often used, but examples of stochastic methods (e.g., [99] where noisy estimates of the derivative are computed instead of the true gradient and the learning rate is annealed), or derivative free non-linear optimization methods (e.g., Powells method [102]) can be found in literature. Determining the globally optimal solution is explored in [2] and [41], but no work, to the best of our knowledge, has considered global optimization for non-rigid image registration. This problem will be explored in Chapter 5.



(a) Target Image. (b) Moving Image. (c) Deformation Field.

Figure 3: An illustration of registration. The output is a deformation field, in Fig. 3(c), that takes the moving image and aligns it with a target image.

Another problem that frequently arises in computer vision is the registration of two point sets. One seeks to obtain a transformation that maps a measured point set to a fixed model set. Point sets are often generated in the field of computer vision; for example, as output of a feature detector applied to an image [37], from higher level user input used to identify meaningful "landmarks" [95], or as output of a scanning device [55]. Once salient point features are extracted, registration on these point sets can be used for a variety of computer vision applications: determining stereo correspondences [84], image set matching [15] for panoramic stitching, or medical imaging [100, 11]. As for image registration, in designing a point set registration, choices for the similarity metric, the representation of the deformation, and the optimization method must be made, which are discussed in Section 4.

Modeling an anatomical structure with a point cloud makes point set registration applicable to medical imagery: the alignment task is simplified because only a subset of the full image data is used, and the computed deformation can be applied for atlas based segmentation. Point sets may be extracted from an image automatically or manually, and point correspondences may be known or unknown. If features are not sufficiently salient, a low-level automatic detection approach will result in poor detection, and manual selection of points may be considered too time-consuming or tedious. Hence, the detection step can be skipped altogether, and image intensities can be matched directly, as discussed above; however, this approach is vastly more computationally demanding because all image data is used, as opposed to a sparse subset of the data.

1.4 Organization of the Thesis and Contributions

The approach for performing registration or segmentation is strongly dependent on the image problem and the availability of prior information such as shape models, an estimate for an object's location, etc. Additionally, for complex images, such as the ones encountered in medicine, user interactivity or at least user constraints are a requirement for achieving satisfactory results. This thesis presents several algorithms that query the user for input and leverage this higher level information to maximally reduce the workload on the human. The contributions of the work in this thesis are:

- **Chapter 1:** Introduces segmentation and registration, explains the link between these tasks, and discusses algorithmic solutions to commonly encountered image scenarios.

- **Chapter 2:** For cases when an object of interest cannot be segmented based on intensity information alone, model based segmentation methods [54] and [93] are powerful, but they require a representative training set during the learning phase. If a training set is not available or cannot be created (e.g., for traumatic brain injury patients due to the uniqueness of each segmentation), the methods cannot be applied.

Thus, a new approach must be taken. In this chapter, an interactive segmentation method is developed that combines an expert user's high level knowledge with an active contour's ability to quickly segment object boundaries, an otherwise tedious task for a person.

- **Chapter 3:** In scenarios where training data is available and shape models can be created, an iterative approach to segmenting multiple objects can be used. We model variations in a single object using PCA and also learn pairwise dependencies of objects. To segment a new object, its location is predicted based on the shape and location of previously completed, correlated structures. This segmentation may not be perfectly accurate, but a correction can be made using semi-automatic active contours or interactive user segmentation, as described in Chapter 2.

- **Chapter 4:** A stochastic approach using particle filtering combined with simulated annealing is presented for point set registration. It is assumed that the points are a sparse representation of a physical object, perhaps created with a rough operation by the user (e.g., global threshold and then sampling of the label mask). Thus, a framework for enforcing injectivity of the deformation is necessary; one is presented for enforcing the non-overlapping and landmark constraints. The method is quantitatively validated on synthetic data and results are shown for point clouds generated from computed tomography (CT) volumes.

- **Chapter 5:** For certain images, intensity information is enough to guide an automatic algorithm. However, another challenge arises in the form of local minima. In this chapter, an image registration technique intended to capture large rigid deformations in addition to large non-rigid deformation components is discussed. The stochastic optimization approach from Chapter 4 is used to capture gross misalignments followed by a gradient descent based registration to compute the finer, remaining errors. Results demonstrate excellent outcomes when applying this method, even for

difficult registration problems.

CHAPTER II

HUMAN SUPERVISORY CONTROL FRAMEWORK FOR INTERACTIVE MEDICAL IMAGE SEGMENTATION

This section presents an interactive approach to segmentation by formulating it as a human supervisory control (HSC) problem. The proposed approach reduces the time required to complete a segmentation (compared to manual segmentation) by leveraging the strength of the user (high level knowledge of anatomy) with the strength of an automatic algorithm (performing an otherwise tedious task of outlining exact boundary locations). This section is based on the work of [51]; it is organized as follows. Section 2.1 introduces variational active contours. In Section 2.2, the sources of difficulty in using global descriptors for segmentation of volumetric medical images are analyzed. A human supervisory control formulation for segmentation, novel in this field, is presented in Section 2.3. Results and comparisons are left for Section 2.4, including a software application developed as part of this work and currently being used in medical research.

2.1 Variational Active Contours

2.1.1 Problem Formulation

Segmentation, as described in Section 1.2, is the task of determining an interface Γ that separates an image domain $\Omega \in \mathbb{R}^n$ into two sub-domains: foreground and background. A variety of methods have been used for solving this problem; this section does not attempt to summarize the vast amount of work in the field but focuses on variational active contours for image segmentation. In the active contour framework, for a two dimensional image, the

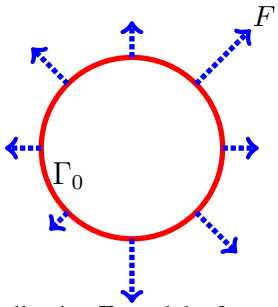
interface Γ is a closed planar curve defined as:

$$\Gamma : I \to \mathbb{R}^2 \text{ where} \tag{1}$$

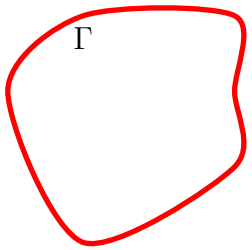
$$I = [a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\} \tag{2}$$

$$\text{s.t. } \Gamma(a) = \Gamma(b) \; . \tag{3}$$

Initialization involves defining a curve Γ_0 as the starting point in the segmentation process; hence, active contours are considered a semi-automatic approach. Now, suppose the curve is allowed to deform according to a force F defined at each point on the curve. These forces move Γ_0 to a desired location, as shown in Fig. 4, and the goal becomes to track the motion of the curve as it evolves.



(a) Initialization Γ_0 and the forces F being applied.



(b) New curve location Γ.

Figure 4: This figure illustrates an initial curve Γ_0 moving according to a force F at each point. Γ is the new location of the curve after these force are applied to deform the curve.

Generally, segmentation is achieved by selecting an energy functional $E(\Gamma)$ and solving the following minimization problem:

$$\min_{\Gamma} E(\Gamma) \; . \tag{4}$$

This optimization problem is solved by finding forces F that move the curve towards the minimum of $E(\Gamma)$. Energy functionals depend on photometric properties of the image and geometric properties of the curve; the geometric properties include any available prior knowledge about the shape model. A number of options for $E(\cdot)$ are presented in Section 2.1.4. To be fruitful, the crucial property any E must exhibit is its global minimum

coincides with the desired segmentation.

A general energy functional of Γ takes the form

$$E(\Gamma) = \int_0^L g\left(\Gamma\left(p\right)\right)\ dp \tag{5}$$

where $g(\cdot)$ is some function of the curve, p is a parameter for the curve, and L is the curve length. The following discussion assumes that a *geometric* energy $E(\cdot)$ is chosen. A *geometric* energy is one that is independent of the curve's parametrization and is composed of quantities that depend only on the geometric properties of the curve. To find the minimum, the first variation of the energy functional is computed, and the curve moves in this steepest descent direction, monotonically, maximally decreasing the energy at each iteration until a local minimum is reached.

The Gâteaux derivative of the energy functional with respect to the curve is computed by introducing a test function v and an artificial time variable t and evaluating

$$\delta E(v) = \left.\frac{d}{dt} E(\Gamma + tv)\right|_{t=0} \quad . \tag{6}$$

Another way of computing the first variation of $E(\cdot)$ is to augment the curve parameter p with a fictitious time variable t to define the family of curves over time as $\Gamma(p,t)$. Then, $E(\cdot)$ is a function of t and the general time derivative is

$$\frac{d}{dt} E(\Gamma(p,t)) = \int_0^L g_t(\Gamma) \cdot \Gamma_t\ dp\ . \tag{7}$$

If the curve dynamics are denoted by $\frac{\partial \Gamma(p,t)}{\partial t}$, motion of the curve according to an arbitrary, time-dependent vector field $\vec{M} : \mathbb{R}^2 \times R^+$ can be decomposed into a tangential and a normal component:

$$\frac{\partial \Gamma}{\partial t} = \vec{M} = \left\langle \vec{M}, \vec{T} \right\rangle \vec{T} + \left\langle \vec{M}, \vec{N} \right\rangle \vec{N} \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, $\vec{T}$ is a vector tangent to the curve, and $\vec{N}$ is the vector normal to the curve. The tangential component only contributes to a re-parametrization without moving the curve. Since we are interested in the motion of the curve, the tangential

component can be discarded. For a *geometric* energy, the curve evolution corresponding to a gradient descent direction is known to take the form

$$\frac{\partial \Gamma}{\partial t} = F\vec{N} \tag{9}$$

where F is a scalar quantity. From Eq. (7), it is clear that decreasing the energy most rapidly is achieved by choosing $F = -g_t(\Gamma)$ and moving the curve in the direction

$$\frac{\partial \Gamma}{\partial t} = -g_t(\Gamma)\vec{N}\ . \tag{10}$$

Thus, the remaining task for implementing a variational active contour model is to propose a way of tracking the curve during its evolution. There are two ways of accomplishing this task: using an *explicit* curve representation, described in Section 2.1.2, or an *implicit* representation, presented in Section 2.1.3.

2.1.2 Parametric Active Contours

The distinction between a parametric representation of active contours and the level set method, discussed in Section 2.1.3, is comparable to using a Eulerian or a Lagrangian specification for a flow field. The Eulerian formulation defines a field at a particular location over time. In the Lagrangian specification, an observer tracks an interface during evolution and the flow for that parcel throughout time is determined. Parametric active contours, or *snakes* are a Lagrangian formulation for tracking dynamic curves [48]. Snakes are represented as parametric curves by dividing the parameter domain into segments and placing marker points accordingly over the length of the curve.

Two common problems are associated with this formulation. First, initially, the marker points uniformly divide the domain; but, as evolution progresses, portions of the curve contain sparsely spaced particles while in other segments, the particle spacing approaches zero. This non-uniform spacing leads to numerical instability. Second, during evolution, the topology defined by the active contour on the image domain may change. For instance, the curve could intersect itself, in which case, the curve should be merged around the

intersection. Also, the curve could shrink towards a point in a neighborhood, in which case it should be split. These changes in topology require sophisticated rules for keeping track of the marker points.

2.1.3 Level Set Method

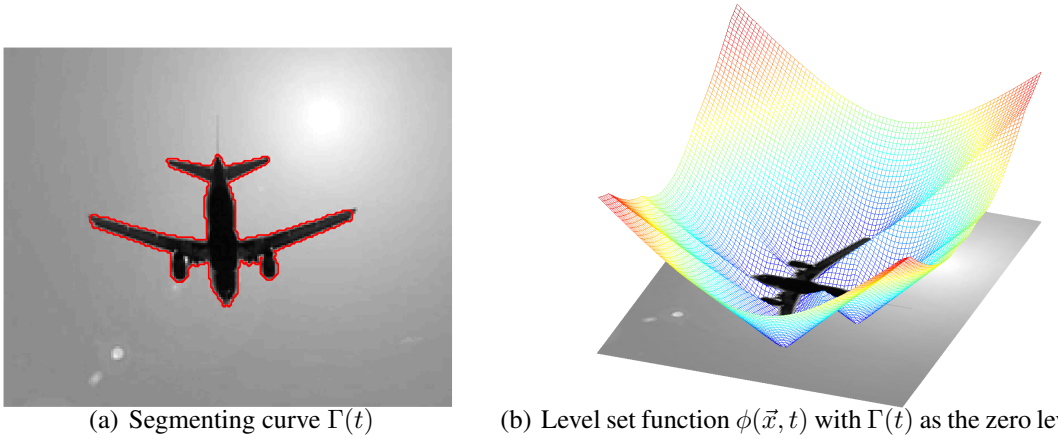
Despite efficient implementations, parametric active contours suffer from an crucial drawback: there is not a principled manner for keeping track of changes in topology. As a solution, the level set method was introduced in the seminal work of Osher and Sethian [68]. In this formulation, the front Γ (i.e., a curve or surface) evolving during segmentation is embedded as an iso-contour of a *level set function* $\phi(\vec{x})$

$$\Gamma(t) = \{\vec{x} \mid \phi(\vec{x}, t) = 0\} \tag{11}$$

whose dimension is one higher than the segmenting interface. $\Gamma(t)$ is the zero level set of ϕ in Eq. (11). Routinely, $\phi(\vec{x}, t)$ is taken to be the signed distance function [67]

$$d(\vec{x}) = \begin{cases} - \min\left(|\vec{x} - \vec{x}_I|\right) & \text{if } \vec{x} \text{ inside } \Gamma \\ \min\left(|\vec{x} - \vec{x}_I|\right) & \text{if } \vec{x} \text{ outside } \Gamma \end{cases} \qquad \forall\, \vec{x}_I \in \Gamma(t) \tag{12}$$

that is negative if $\vec{x}$ is inside the curve and positive if $\vec{x}$ is outside of the curve. A sample Γ and the corresponding ϕ are shown in Fig. 5.



(a) Segmenting curve $\Gamma(t)$ (b) Level set function $\phi(\vec{x}, t)$ with $\Gamma(t)$ as the zero level set.

Figure 5: A sample image and a segmenting curve (red) at time t are show in Fig. 5(a). The corresponding signed distance function is displayed as a surface in Fig. 5(b).

This representation was introduced as a convenient way to keep track of Γ. As presented in Section 2.1.1, an active contour moves in the normal direction according to some force F. Thus, a dynamics equation for $\phi(\vec{x}, t)$ is required that will allow the level set function to match the motion of Γ, preserving $\Gamma(t)$ as its zero level set at all times. By definition,

$$\phi(\Gamma(t), t) = 0 \tag{13}$$

and taking a time derivative of Eq. (13) yields

$$\frac{\partial}{\partial t}\left(\phi\left(\Gamma\left(p,t\right),t\right)\right) = \phi_t + \nabla\phi\left(\Gamma,t\right)\cdot\frac{\partial\Gamma}{\partial t} = 0\,. \tag{14}$$

The normal vector is $\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}$, which leads to the level set equation

$$\phi_t + F|\nabla\phi| = 0 \tag{15}$$

$$\text{given } \phi(\Gamma, 0) = \Gamma_0 \tag{16}$$

where $F = \frac{\partial\Gamma}{\partial t}\cdot\vec{N}$, the force applied in the normal direction to Γ, and Γ_0 is an initialization for the interface. Several choices for F are presented in the next section.

Several methods for speeding up the computations above have been proposed. First, the Fast Marching Method [85] is used to solve a particular case of the Eikonal equation

$$|\nabla\phi(x)| = 1 \text{ subject to} \tag{17}$$

$$\phi(x)|_{\Gamma} = 0 \tag{18}$$

for initialization of ϕ to the signed distance function, efficiently. Second, it should be noted that for segmentation, only motion of the zero level set of ϕ is of interest, and computation for ϕ over the entire domain are unnecessary. Further, Eq. (14) defines the dynamics of ϕ at the zero level set only; extensions to level sets away from Γ have to be defined alternatively. A narrowband method [105] involves initializing and propagating ϕ in a narrow region around the segmentation interface. This approach is clearly more computationally efficient and avoids extrapolating propagation speeds far away from Γ.

2.1.4 Image Based Energy Functionals

In Section 2.1.1-2.1.3, tracking an active contour during evolution and choosing curve dynamics that move an initial contour to a minimum of a chosen functional was discussed. A number of energies have been proposed in the literature for segmentation. For instance, conformal (geodesic) active contours were introduced in [49, 17]:

$$E_g(\Gamma) = \int_0^1 g(|\nabla I(\Gamma)|)\, |\Gamma^{'}(p)|\, dp \quad \text{where} \tag{19}$$

$$g(|\nabla I|) = \frac{1}{1 + |\nabla \hat{I}|^p}\,, \tag{20}$$

$p = 1$ or 2 or $g(\cdot)$ is another edge detecting function and $\hat{I}$ is a smoothed version of the image. This energy is applicable to objects with strong intensity gradients corresponding to salient edges and is designed to have a minimum when the segmenting curve is located on the object boundary. However, an object may not have distinct boundaries or an image may be corrupted by noise, which produces frequent, spurious edges. Therefore, edge-based methods are likely to fail, and a region based energy is preferred.

Chan and Vese [19] explores this scenario and defines an energy that is minimal when the contour maximally separates the mean intensity inside and outside of the contour:

$$E_m(\Gamma) = \int_{inside(\Gamma)} |I(x,y) - c_{in}|\, dx\, dy + \int_{outside(\Gamma)} |I(x,y) - c_{out}|\, dx\, dy \tag{21}$$

where c_in, c_out are constants. A correct segmentation can be achieved using the Chan-Vese energy when a homogeneous object is captured on a homogeneous background of different intensity. If the first moment of an object's intensity distribution is the same mean the background pixels, E_m is insufficient to discriminate between the two.

Another region-based approach was proposed in [63]; it works by maximally separating entire intensity distributions between the object and background based on the distributions'

Bhattacharyya distance. The Bhattacharyya segmentation energy is

$$E_b(\phi) = \int_{\mathbf{z}} \sqrt{P_{in}(\mathbf{z}|\phi(x))P_{out}(\mathbf{z}|\phi(x))}\, dz \qquad \text{where} \tag{22}$$

$$P_{in}(\mathbf{z}|\phi(x)) = \frac{\int_\Omega K_{in}(\mathbf{z} - J(x))\mathcal{H}(-\phi(x))\, dx}{\int_\Omega \mathcal{H}(-\phi(x))\, dx} \tag{23}$$

$$P_{out}(\mathbf{z}|\phi(x)) = \frac{\int_\Omega K_{out}(\mathbf{z} - J(x))\mathcal{H}(\phi(x))\, dx}{\int_\Omega \mathcal{H}(\phi(x))\, dx} \tag{24}$$

$$\mathcal{H}(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases} \tag{25}$$

and $J(x)$ is a vector valued feature image obtained from the original image $I(x)$. Hence, an object with a first moment similar to the background can be distinguished by considering higher order image statistics.

Frequently, an object of interest and the background are not homogeneous spatially, and while the intensity profiles are discriminative locally, at the global scale the same descriptors are insufficient for segmentation. A local active contour [52] was proposed to segment these cases:

$$E_l(\phi) = \int_{\Omega_x} \delta\phi(x) \int_{\Omega_y} F(\phi(y), I) dy\, dx \qquad \text{where, for example} \tag{26}$$

$$F = \mathcal{B}(x,y) \cdot \left(\mathcal{H}(\phi(y))\, (I(y) - u_l)^2 + (1 - \mathcal{H}(\phi(y)))\, (I(y) - v_l)^2 \right) \tag{27}$$

$$u_l = \frac{\int_{\Omega_y} \mathcal{H}\left(\phi(y)\right) \cdot I(y) dy}{\int_{\Omega_y} \mathcal{H}\left(\phi(y)\right) dy} \tag{28}$$

$$v_l = \frac{\int_{\Omega_y} \left(1 - \mathcal{H}\left(\phi\left(y\right)\right)\right) \cdot I(y) dy}{\int_{\Omega_y} \left(1 - \mathcal{H}\left(\phi\left(y\right)\right)\right) dy} \tag{29}$$

$$\mathcal{B}(x,y) = \begin{cases} 1 & \text{if} \quad ||x - y|| < r \\ 0 & \text{if} \quad \text{other} \end{cases} \tag{30}$$

and $\mathcal{B}(x,y)$ determines the restricted domain in which image properties are computed. The energy is chosen depending on the given segmentation scenario.

2.2 *Control Formulation for Image Segmentation*

2.2.1 Problem Analysis

In segmentation, it is rare that an homogeneous object is being segmented in front of a homogeneous background of different intensity. Thus, it becomes difficult to define an energy whose minimum coincides with the object boundaries. Additionally, a shape prior may not be available. An example of this scenario is segmentation of bony tissues in a magnetic resonance imaging (MRI) images. Intensity values in computed tomography (CT) images are in Hounsfield units and their correspondence to various tissues in the body is known. Hence, segmentation can be accomplished by simply filtering out the desired intensities. On the other hand, in MRI, intensities are relative and do not correspond to the same tissue for different patients [104]. Additionally, there is large variance in intensities that correspond to the same organ. In Fig. 6 and 7, we show the degree of intensity inhomogeneity within and between slices, respectively, of the femur. From this graphic, it is clearly impossible to find a single narrow range of intensities that would separate the bone from other organs; also, the intensities that describe bone are spatially varying.

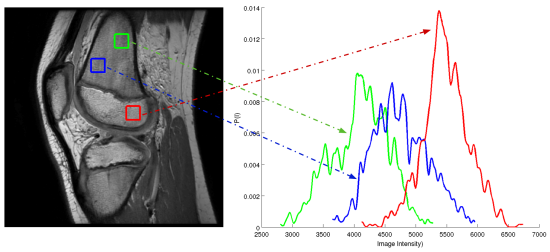


Figure 6: Bone tissue within one MRI slice is highly inhomogeneous.

2.2.2 User Control Framework

A control system is a mechanism (manual or automatic) that regulates the behavior of a plant; a plant is a combination of a process and an actuator. For instance, the process being controlled in this work is the variational active contour segmentation. In existing

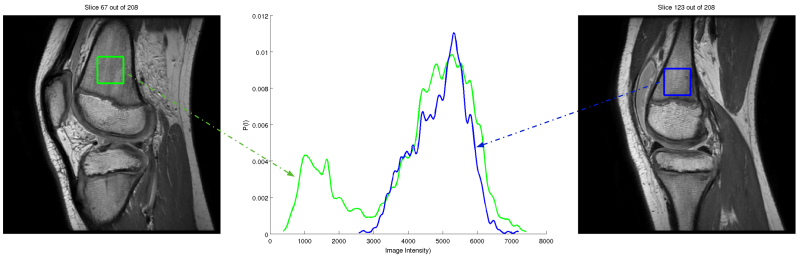


Figure 7: Inhomogeneity of bone tissue across MRI slices is significant.

applications, the segmentation is automatic once initialized by the user. A control loop involves computing the error between the desired reference signal ϕ_{ref} and the current plant output ϕ_{curr} and feeding it back to the controller, which computes a control signal u designed to drive the plant output towards the reference signal.

Human supervisory control (HSC) describes the interaction of an automatic process or task with a user who periodically guides it towards a particular set-point by providing additional input [86]. The automatic process is monitored by the user and corrected if it deviates from the desired behavior. An example from medical imaging is [60] where the authors created a graphical user interface (GUI) and used it to help position a model of a knee implant. User input is employed to perturb the automatic registration algorithm when it becomes trapped in a shallow local minimum or selects a minimum that is bio-mechanically incorrect.

2.3 Interactive Image Segmentation

2.3.1 General HSC Segmentation Framework

An HSC framework is presented for performing segmentation of medical images. The entire process is shown graphically in Fig. 9. Generally, the region-based active contour

segmentation problem can be written as:

$$\min_{\phi} \quad E(\phi) = \int_{\Omega} g(\phi, I)dx + \lambda \int_{\Omega} ||\nabla \mathcal{H}(\phi)||^2 dx \tag{31}$$

$$\mathcal{H}(x) = \begin{cases} 1 & \text{if} \quad x > \quad \epsilon \\ 0 & \text{if} \quad x < - \quad \epsilon \\ \frac{1}{2}\left(1 + \frac{x}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi x}{\epsilon})\right) & \text{otherwise} \end{cases} \tag{32}$$

$$\delta(x) = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{if} \quad |x| < \epsilon \\ \frac{1}{2\epsilon}\left(1 + \cos(\frac{\pi x}{\epsilon})\right) & \text{otherwise .} \end{cases} \tag{33}$$

Here, ϕ is the level set function, I the image data, λ the curvature penalty, and ϵ is a scalar that determines the smoothness of the approximation to the Heaviside function, $\mathcal{H}$, and the approximation to the Dirac delta function, δ. The evolution equation for the level set function is:

$$\phi_t = \delta(\phi)\left(-g_\phi(\phi, I) + \lambda\kappa\right) = G(\phi, I) \tag{34}$$

$$\kappa = -\nabla \cdot \frac{\nabla\phi}{||\nabla\phi||} \quad . \tag{35}$$

where κ in Eq. (35) is the curvature.

Suppose the zero level set of some ϕ^* is the ground truth segmentation result. Then, a "good" function $g(\phi, I)$ is one that causes ϕ to converge to ϕ^* when initialized within some small region around ϕ^* (i.e., ϕ^* is a local minimum of $E(\phi)$). In other words, a $g(\phi, I)$ is "good" if there is some ϕ_0 that converges to ϕ^* and for some positive function ϵ satisfying

$$\phi^*(\vec{x}) - \epsilon(\vec{x}) \leq \phi_0(\vec{x}) \leq \phi^*(\vec{x}) + \epsilon(\vec{x}) \tag{36}$$

with ϵ defining the region of convergence. Unfortunately, both ϕ and ϵ are unknown and the function $g(\phi, I)$ may have certain exceptional regions where it is not discriminative between object and background. To remedy these problems, the user is present in the segmentation process and guides the automatic algorithm. His function is to provide intermittent input to change ϕ. It is assumed that when the user is shown a slice from a 3D

medical image sequence, he/she can generate the correct segmentation ϕ^* (i.e., the user knows the ground truth and given enough time could outline the ideal segmentation result).

As illustrated in Fig. 8, during segmentation, the user provides a finite number of inputs at times t_i for $i \in 1, ..., k$. His input influences the dynamics of $\phi(\vec{x}, t)$. The function modeling the cumulative user input $U(\vec{x}, t)$ up to time t is written as

$$g_k(\vec{x}) = \phi(\vec{x}, {t_k}^+) - \phi(\vec{x}, {t_k}^-) \tag{37}$$

$$u(\vec{x}, t) = \sum_{i=1}^{k} g_k(\vec{x})\delta(t - t_i) \tag{38}$$

$$U(\vec{x}, t) = \int_0^t u(\vec{x}, \tau)d\tau \quad . \tag{39}$$

The aim for the HSC algorithm is to perform segmentation while minimizing the amount of user input, i.e., reduce the domain over which $U(\vec{x}, t)$ is actuated and $||U||_{L_2}$.

Within each time interval $[t_m, t_{m+1}]$, $m \in 1, ..., k-1$ (refer to Fig. 8) during segmentation, it is assumed the user is satisfied with the performance of the automatic algorithm because no input is provided as guidance. It is important to note that simply alternating between manual labeling and automatic segmentation does not produce correct behavior: user input would be overridden during the automatic phase. Instead, it is desirable to incorporate user input into the system dynamics. Then, Eq. (34) takes the form:

$$\phi_t = \underbrace{G(\phi, I)}_{\text{nominal}} + \underbrace{H(\phi, U)}_{\text{control}} \ . \tag{40}$$

2.3.2 Selecting an HSC Segmentation Energy

From the discussion of Section 2.2.1, we recognize that the energy used for segmentation must be local since the organ (in our case, bone) has statistics that change drastically spatially. While it is true that intensities of an organ's tissue do not fall in a narrow range globally, locally the bone appears approximately homogeneous with a clear separation between the object of interest and the background; hence, we select a local variant of the

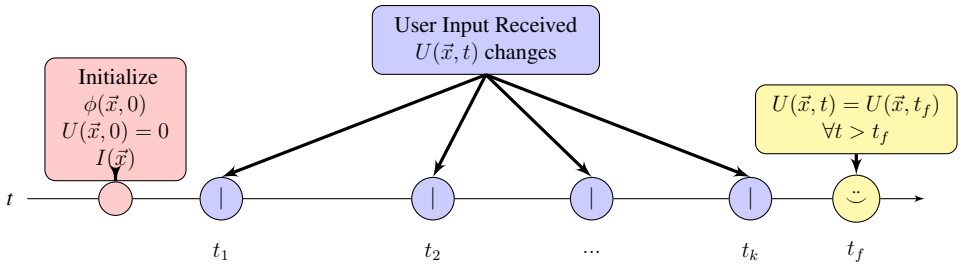


Figure 8: Within each interval $[t_m, t_{m+1}]$, the level set evolves automatically (the plant in Fig. 9). Then, at each time t_{m+1}, user input is provided (the controller in Fig. 9) accumulating in $U(\vec{x},t)$. At t_f, the user determines that $\phi(\vec{x},t)$ is satisfactory and segmentation is complete.

Chan-Vese energy, $E_l(\phi)$ from Eq. (26), as presented in [52] and segment based on first moments. An added benefit of using this local energy is that user input in one region will have no effect on a remote region of the contour, which is intuitive for the user. The functional presented in [52] has been modified to incorporate the novel HSC segmentation term (the 2^{nd} integral in the functional), which makes the dynamics of the evolving curve depend on the user input as it is being provided interactively. The level set formulation is presented in Eq. (41)

$$\min_{\phi} \; E_{cv}(\phi) = E_l(\phi) + \frac{K_U}{2}\int_{\Omega}(\phi - U)^2 U^2 dx + \lambda \int_{\Omega} ||\nabla \mathcal{H}(\phi)||^2 dx \; . \tag{41}$$

Here, ϕ is the level set function, K_U is the gain (scalar) on the user control term, U from Eq. (39) is the integral of the user input, λ is the curvature penalty term, and $\mathcal{H}(x)$ and $\delta(x)$, defined in Eq. (32)-Eq. (33), are smooth approximations to the Heaviside and the Dirac delta functions, respectively. Using the calculus of variations, the dynamics equation, Eq. (42), to minimize the functional in Eq. (41) is:

$$\phi_t = G(\phi, I) + H(\phi, U) \tag{42}$$

$$G = \delta\phi(x)\left(\int_{\Omega_y} \mathcal{B}(x,y)\delta\phi(y)\cdot((I(y)-u_l)^2 - (I(y)-v_l)^2)dy + \lambda\kappa\right) \tag{43}$$

$$H = K_U(\phi(x) - U(x))\cdot U^2(x) \; . \tag{44}$$

In Eq. (36), $\epsilon(\vec{x})$ is the spatially varying width of the region of convergence from the

ground truth segmentation. Its dependence on spatial location means that in certain places ϕ is tolerant of large perturbations while in others care must be taken (i.e., increased user input is required) to drive ϕ towards ϕ^* and assure convergence. Selecting a local active contour energy functional allows the user to concentrate input on a particularly sensitive region of ϕ without worrying about global effects.

2.3.3 User Interaction

In this section, we describe in detail how the user participates in the feedback loop shown in Fig. 9. First, the user provides a rough contour initialization and allows the contour to evolve for a set time Δt. In real time, the result is computed and displayed. Then, as shown in the bottom loop in Fig. 9, the user provides input only in regions where the contour did not appear to move towards the object boundaries; this modified contour $\phi(x, t_k^+)$ serves as the initialization at the next time step $t + \Delta t$ and the region of user input is recorded. The user compares visually his knowledge of the desired segmentation $\phi^*(\vec{x})$ to the current segmentation $\phi(\vec{x}, t)$. If there are errors, three events can explain them: some regions of ϕ_0 were outside of the interval in Eq. (36), the time Δt was too short for the level set to converge, or the nominal dynamics are insufficient to drive the segmenting interface to the desired location.

After each time interval of Δt, the automatic algorithm returns a segmentation for visualization, enabling the user to optionally generate another of pulse function g_k in Eq. (38) or simply continue the segmentation. Further inputs by the user occur only in places where corrections are desired and the input is accumulated in $U(\vec{x}, t)$; in this way, the algorithm learns regions where significant input has been provided over time (i.e. the energy in Eq. (41) is not discriminative in these regions of the image and user input should dominate here). A detailed convergence analysis of this approach is presented in [46]. Currently, segmentation is done slice by slice for the image volume although the same approach can be extended to 3D. Consistency between slices in maintained by the user. Once slice k is

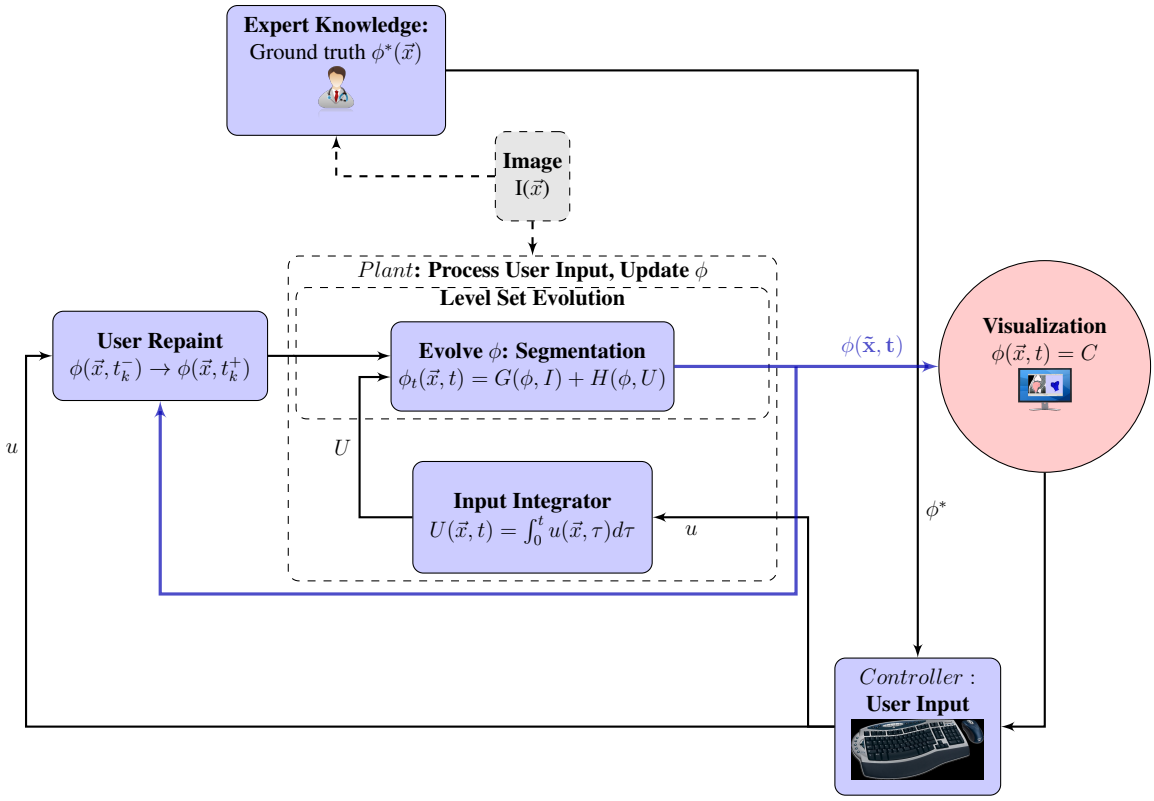


Figure 9: Visualization feedback to the user allows him to supervise the automatic segmentation and exploits his *expert knowledge* as prior information for the algorithm. This "human in the loop" structure compensates for a poor initialization or sub-optimal choice of segmentation energy (the appropriateness of both is typically difficult to judge *a priori*) for a particular scene.

completed, a "copy and paste" operation is performed to transfer this 2D label map onto slice $k + 1$; it serves as the initialization for the segmentation of slice $k + 1$.

2.4 Results & Analysis

2.4.1 Results of HSC Interactive Segmentor

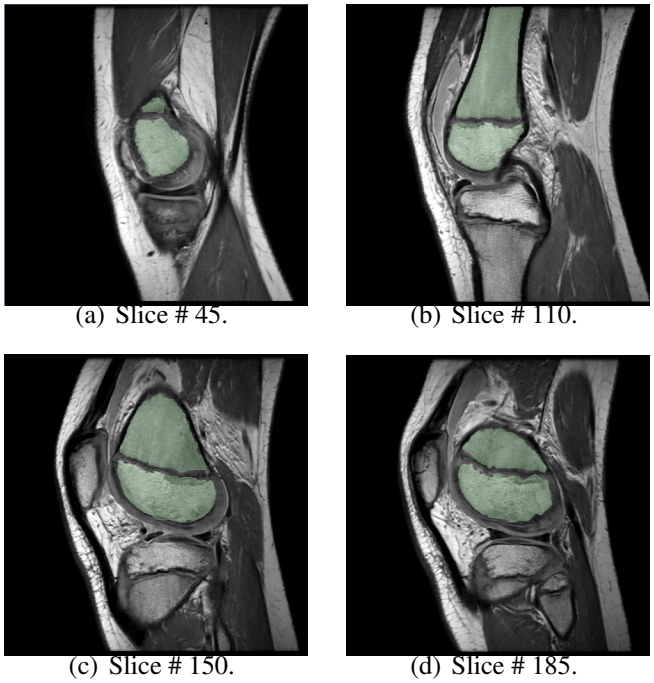
Data in this study consists of approximately 300 high-resolution (512x512x200) MRI scans of the knee that are being segmented to obtain 3D models of the metaphysis and epiphysis. Shape analysis will be performed on these segmentations to model bone age and growth potential. An interactive segmentation approach must be evaluated based on three criteria: the time it requires to segment a case, the accuracy of the segmentation, and the ease of use. A typical segmentation result with the method from Section 2.2.2 and 2.3.3 is shown

in Fig. 10.

2.4.2 Comparison to Semi-Automatic Segmentation Methods

Recent work on semi-automatic segmentation consists primarily of methods based either on *graphs* [14, 12] or *active-contours* [68]. Often these methods are sensitive to initialization and it is not a simple task to provide a good initial label volume. As seen in Fig. 9, the output of our algorithm converges to the ground truth; for our comparisons we initialize using the ground truth because this is the best case scenario. The final result of these methods is worse than the initialization.

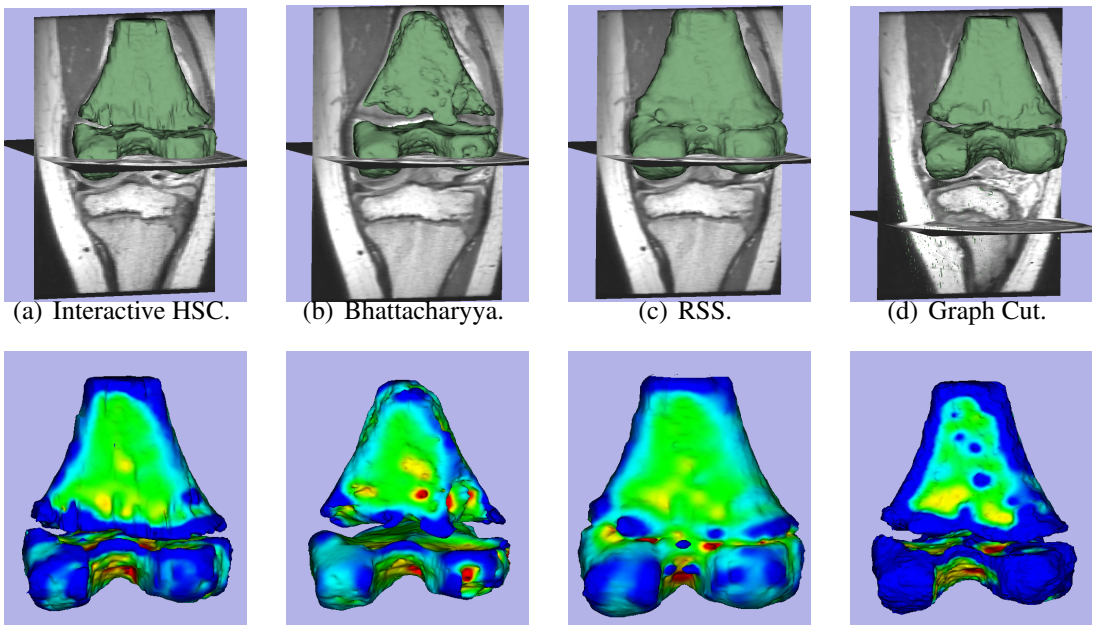
We present results using Bhattacharyya [63] and RSS [32] active contours and Graph Cuts. In Fig. 11, the top row of models are the segmentation results and the bottom row are figures showing curvature of the corresponding model surface in the top row. Bhattacharyya segmentation separates intensity distributions, but in our case background



(a) Slice # 45. (b) Slice # 110.

(c) Slice # 150. (d) Slice # 185.

Figure 10: This figure shows result achieved using the HSC Interactive Segmentor. By zooming in on the images, it becomes noticeable that this algorithm attains accurate segmentation results even for complex image regions (around the growth plate).



(a) Interactive HSC. (b) Bhattacharyya. (c) RSS. (d) Graph Cut.

Figure 11: The results of our algorithm are in 11(a); it is the ground truth. Models in 11(b)-11(d) are results of Bhattacharyya, RSS, and Graph Cuts, respectively. These algorithms were initialized with the ground truth from 11(a). In the bottom row, surface curvature is displayed; these models were computed on a mesh generated from the corresponding models(by column) in the first row. One should notice that even initialized with ground truth, the algorithms in 11(b)-11(d) move away from the correct result; this means that all of these energies do not have the ground truth as a local minimum for any region. Also notice (as more clearly shown by the surface curvature) that the largest errors using these energies occur in regions of greatest importance for us: around the growth plate separating the two pieces of bone.

and foreground distributions are overlapping, which leads to the under-segmentation in Fig. 11(a). RSS performs segmentation using robust statistics like median and inter-quartile range but these descriptors are insufficient as seen in the over-segmentation of the organ in Fig. 11(c). Finally, in Fig. 11(d), it can be seen that numerous small islands form far away from the initialization because Graph Cuts performs global optimization and the largest errors are near boundaries, which are the most time consuming for a human.

Model based segmentation methods [54] and [93] are powerful but they require a representative training set during the learning phase. Prior to the completion of our population study, a training set was not available. For future work we plan to incorporate shape priors into our interactive segmentation framework.

2.4.3 Comparison to Interactive Segmentation Methods

The live-wire algorithm is an interactive segmentation method that "snaps" to image edges [8]. In [38], the user provides control points by segmenting several slices, with the help of the live-wire algorithm, and variational interpolation is performed on the resulting cloud of points. In [1], "Spotlight" guides the user to regions that require additional input; input is provided in a plane with the live-wire algorithm. Our approach is in fact complementary to these works since we concentrate on segmentation within one volume slice and they emphasize interpolating between slices.

ITK Snap [106] provides tools to manually perform segmentation and to initialize and view the results of active contour. This application is excellent for semi-automatic segmentation. In this chapter, segmentation is done interactively: the described approach tightly couples user input with an automatic process throughout the segmentation. In [18], the authors implement a GPU based level set solver to make interaction real time. However, they focus on interactively modifying parameters used in the energy functional, which is likely to be insufficient for obtain the desired result.

For timing comparison, several interactive methods were tested; segmentation of 5 MRI cases was performed and the average times are presented. Each scan had approximately 200 slices of which 150 contained bone. Manual segmentation took 90 min, 35 sec per slice. Segmenting with the live-wire algorithm, one slice took 20 sec, or 50 min per case. With the HSC Segmentor, an MRI case was completed in 10 min, or 6.5 sec per slice and segmentation is of manual quality.

2.5 Chapter Conclusion

In this chapter, interactive segmentation was phrased as a human supervisory control task; there is a tight coupling between user input and automatic segmentation. Consequently strengths of manual and automatic segmentations are leveraged. The user acts as a supervisor by providing high level guidance in the form of a close initialization to an automatic

algorithm, based on variational active contours. User input is occasional and local; it iterates with the automatic portion which excels at discriminating between object and background if provided a good initialization. It is quick for a user to roughly outline an object of interest but tedious to outline the boundaries exactly, which is a strength of the automatic segmentation tools. The work presented in this chapter is clinically relevant and is being used for a population study of human skeletal growth. Results of the approach show that a segmentation that is qualitatively comparable to manual segmentation results can be achieved in less than $\frac{1}{6}$ of the time. Also we demonstrate a user interface that is simple, cross-platform, light-weight and intuitive.

CHAPTER III

COUPLED SEGMENTATION FOR ANATOMICAL STRUCTURES BY COMBINING SHAPE AND RELATIONAL SPATIAL INFORMATION

The volume of data from computed tomography (CT) or magnetic resonance imaging (MRI) is often prohibitively large to process manually; consequently, certain steps of treatment planning, deemed not critical by doctors, are skipped, which leads to a sub-optimal treatment. For example, external-beam radiation therapy is used to provide highly accurate doses of radiation to patients undergoing cancer treatment. The most common form is fractionated delivery, in which a patient receives therapy for up to eight weeks; it is crucial to deliver precise doses to cover the target fully while reducing damage to healthy cells and to correlate radiation doses across sessions. To maintain consistency across a fractionated therapy plan, the deformation and precise location of the target and critical organs needs to be known.

In this application, critical structures need to be segmented, and the number of targets can be in the teens. Segmentation based on intensities alone tends to "leak" into surrounding tissue if unconstrained by some form of prior knowledge. Shape models have been used to provide prior knowledge for a segmentation algorithm but there is no coupling between multiple segmentation targets in these approaches. Medical images can be corrupted by various artifacts which complicate the problem of automatic segmentation. Photon starvation can cause streak artifacts in CT, especially in the areas where metal prostheses have been introduced. Furthermore, when the object of interest is a zone rather than an organ, such as with cervical lymph nodes, there is a danger in allowing the image gradient to influence the shape prior.

In this chapter, a method is introduced to segment a sequence of related objects; shape models are built to capture relative shape and spatial information between structures. A statistical technique called canonical correlation analysis (CCA) [40] is used to relate two sets of variables. This method was applied in [75] to predict depth maps of features from color images of faces. Using a probabilistic canonical correlation analysis (PCCA) [7], we make a model that determines the location of an anatomical structure given the location of previously segmented structures. Segmentation is performed sequentially with results for completed structures serving as landmarks to build a prediction for the shape and location of future structures. This prediction is the shape prior used by a segmentation algorithm.

3.1 Motivating Problem

There are two main scenarios that lead to poor results of a (semi)automatic segmentation algorithm on the CT images used in adaptive radiotherapy. The first is unwanted artifacts whose effect is demonstrated in a slice of a CT scan in Fig. 12; the star-shaped pattern of artifacts seen here is caused by dental fillings in the patient's mouth. The image intensity of the artifacts is similar to the nearby bony structure called the mandible, which is a structure

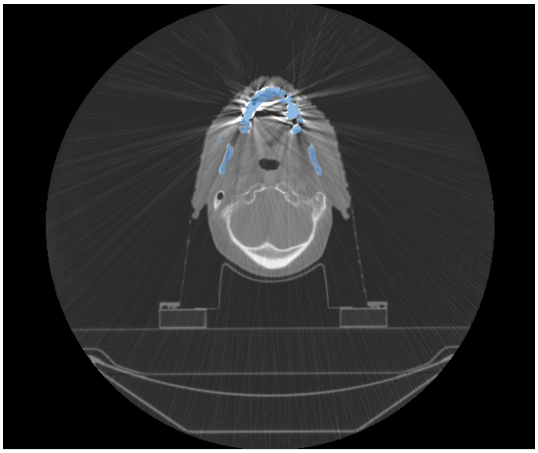
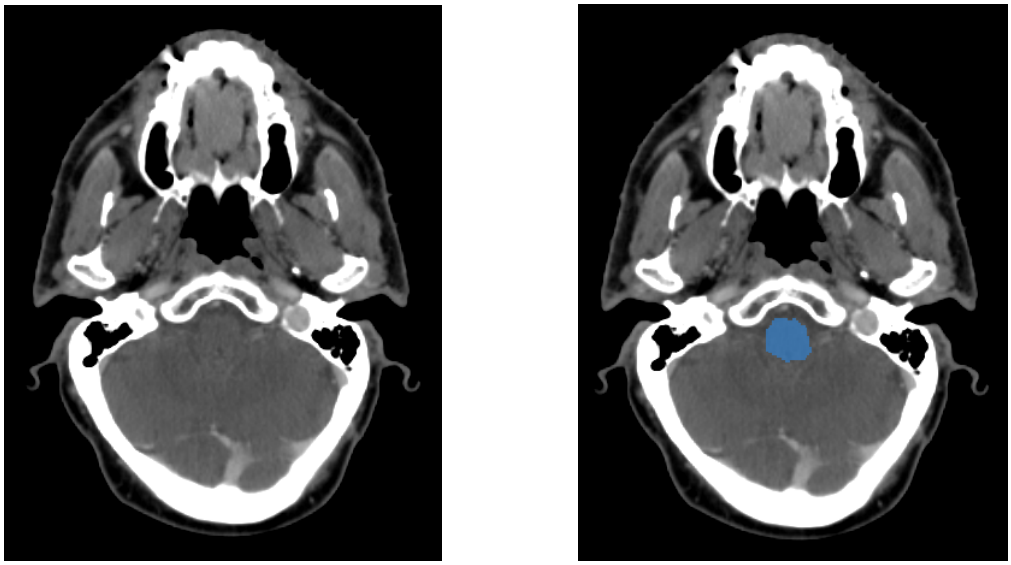


Figure 12: CT slice with artifacts from dental fillings. The blue label is the desired segmentation result, manually drawn by a physician. It is clear that the artifacts complicate the segmentation problem.

of interest. Since these artifacts are present in multiple slices of the CT scan, a 3D segmentation algorithm tends to make errors in segmentation by capturing portions of the artifacts as part of the structure. The other sources of difficulty are organs similar in intensity to neighboring tissue. In Fig. 13(a), a slice of a CT scan is shown with the brain stem present; the ground truth segmentation, made manually by a physician, can be seen in Figure 13(b).

A clinician is able to perform manual segmentation of an anatomical structure such as the brain stem in Fig. 13 or a mandible occluded by image artifacts in Figure 12 by using other structures as landmarks and his knowledge of spatial relationships between them. This chapter formalizes these spatial dependencies and uses them to make a prediction for the likely location of a target structure given structures with completed segmentations.



(a) CT slice with the brain stem present.

(b) Same CT slice in Fig. 13(a) with the ground truth segmentation superimposed on top.

Figure 13: This figure shows a sample structure, the brain stem, that has soft boundaries and a small difference in intensity between surrounding tissue. A segmentation algorithm without a shape prior exhibits poor performance in this situation by leaking into neighboring brain tissue.

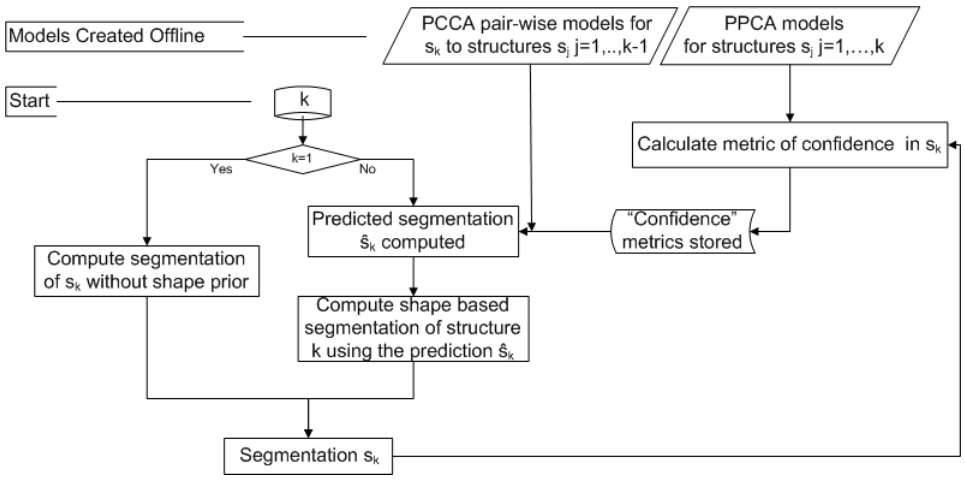


Figure 14: Visual description of the approach presented in this chapter. $k \in 1, ..., N$ is the structure index.

3.2 Proposed Framework

First, a graphical view of the overall proposed framework is provided in Fig. 14 and then the building blocks are explained in detail. It is assumed a physician has a list containing anatomical structures of interest: $\tilde{s}_k \ \forall \ k \in 1, ..., N$. Then, he qualitatively ranks and re-labels them as $s_k \ \forall \ k \in 1, ..., N$ based on the difficulty of segmenting each one with intensity information alone: s_1 has the clearest boundaries and s_N has the least distinctive boundaries. Offline, a probabilistic principle component analysis (PPCA) [91] is performed on manually generated segmentations of organs to create a model for each anatomical structure s_k , $k \in 1, ..., N$. The model contains the average value and typical variances of the structure across the human population. Also offline, PCCA is used to train a model that, given an observation of structure s_j, will compute the most likely structure s_k; so, for each structure s_k, there are $k-1$ PCCA models to relate s_k with each preceding s_j.

The algorithm is initialized by segmenting s_1 without any shape priors with an approach of the user's choice; it is expected that an accurate result can be quickly obtained because s_1 is the least problematic structure to segment. Then, using the PPCA model created offline, a metric of confidence is computed in the observed segmentation and denoted by p_1. At the k^{th} stage of the algorithm, segmentations for structures s_i along with confidence metrics

p_i for $i \in 1, ..., k-1$ are available. For each structure s_i, a point $s_{k|i}$ is computed, which is the predicted segmentation of s_k given the completed segmentation s_i. Consequently, there are $(k-1)$ predictions for $s_k : \{s_{k|1}, ..., s_{k|k-1}\}$ and $(k-1)$ corresponding confidence metrics, one for each prediction. Then, these predictions are fused to form a single, best prediction for s_k. Finally, model based segmentation is performed using the prediction for s_k and the result is fed back into the algorithm to serve as prior information for segmenting s_{k+1} . Section 3.2.1 discusses PPCA used to compute p_i and section 3.2.2 explains how to perform PCCA to compute the predictions $s_{k|j}$.

From an image volume containing $M \times N \times P$ voxels, the segmentation v_b is represented as a binary volume of the same size with a value of '1' at voxel locations that are part of the target structure and a value '0' at locations that are not. The volume v_b is re-arranged into a vector by moving from slice 1 to P, extracting the columns in order from each slice, and stacking them vertically. This vector is called s, $s \in \mathbb{R}^D$ where $D = MNP$; it is the most convenient way to represent a segmentation for this work. The objective is to find this vector s_i for $i \in [1, .., N]$, as would be determined by a physician. In this chapter, the goal is precisely to build a model that can be used as a prior for segmentation. To this end, we use the structures $s_1, ..., s_{k-1}$ as landmarks and form a prediction, which is denoted $\hat{s}_k$, for the likely value of s_k, the segmentation of the current structure. The prediction is made by using the following model

$$\hat{s}_k = p_{k-1} s_{k|k-1} + \cdots + p_1 s_{k|1} \tag{45}$$

to fuse the information from each segmentation, which is a weighted combination of all predictions.

3.2.1 Computing the Confidence Metric

At stage k, the binary vectors s_i for $i \in [1, \cdots, k]$ are available (i.e., k targets have been segmented). Likelihood of obtaining these segmentations must be computed to have a

measure of confidence in the segmentations' accuracy. For each target structure, PPCA is performed. The results of this analysis is the average value of s_k and typical variations present in the population; from this information, the probability of drawing s_k is computed. PCA is typically phrased as finding a linear projection of the samples $\{x_1, ..., x_n\}$ with $x_k \in \mathbb{R}^D$ onto an $L < D$ dimensional subspace that maximizes the variance between the components of the projection. PPCA, instead, takes a generative view of the problem as in Fig. 15

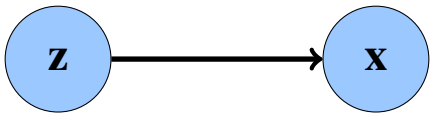


Figure 15: Generative model for PPCA.

and introduces a latent variable z, which represents the lower dimensional probabilistic principal components subspace. The observed variable is expressed in terms of a latent variable projected onto the original $D-$dimensional subspace, the mean of the observed variable, and the measurement noise:

$$x = Wz + \mu + \epsilon \tag{46}$$

In (46), z is the $L-$dimensional latent variable, x is the $D-$dimensional observed variable, μ is the mean of the observed variable, and ϵ is the measurement noise with a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. Hence, the PPCA model may be written as:

$$z \sim \mathcal{N}(0, I) \tag{47}$$

$$x|z \sim \mathcal{N}(Wz + \mu, \sigma^2 I) \tag{48}$$

$$z|x \sim \mathcal{N}(M^{-1}W^T(x - \mu), \sigma^2 M^{-1}) \tag{49}$$

$$M = W^T W + \sigma^2 I \tag{50}$$

$$W = U_L(\Lambda_L - \sigma^2 I)^{\frac{1}{2}} R \tag{51}$$

$$\sigma^2 = \frac{1}{D-L}\sum_{i=L+1}^{D} \lambda_i \quad . \tag{52}$$

Given K samples x of dimension D, the mean μ is equal to the sample mean

$$\mu = \hat{\mu} = \frac{1}{K}\sum_{j=1}^{K} x_j \tag{53}$$

and the covariance matrix is equal to the sample covariance matrix, which is defined as

$$\Sigma = \hat{\Sigma} = \frac{1}{K}\sum_{j=1}^{K} (x_j - \mu)(x_j - \mu)^T \ . \tag{54}$$

A symmetric matrix can always be decomposed as $\Sigma = U\Lambda U^T$. Λ is a diagonal matrix with the eigenvalues of Σ on the diagonal in decreasing order, and U is an orthogonal matrix whose columns are the corresponding eigenvectors. The matrices U_L and Λ_L use the largest L eigenvalues and the corresponding L eigenvectors. In Eq. (52), the λ_i are the eigenvalues of Σ, and R in Eq. (51) is a rotation matrix, typically chosen $R = I$. The equations for W and σ^2 are derived in [91] using maximum likelihood. A note: as $\sigma^2 \to 0$ the probabilistic model becomes equivalent to the standard PCA model.

Since the mean, median, and mode all take the same value for a normal distribution, given an x, the most likely latent variable is

$$z^* = \mathbb{E}\left[z|x\right] = M^{-1}W^T(x - \mu) \quad . \tag{55}$$

Similarly, Eq. (56) allows the most likely observed variable to recovered given a z:

$$x^* = \mathbb{E}\left[x|z\right] = Wz + \mu \quad . \tag{56}$$

The Mahalanobis or the generalized squared distance is defined as

$$D(y) = \sqrt{(y - \mu)^T \hat{\Sigma}^{-1}(y - \mu)} \tag{57}$$

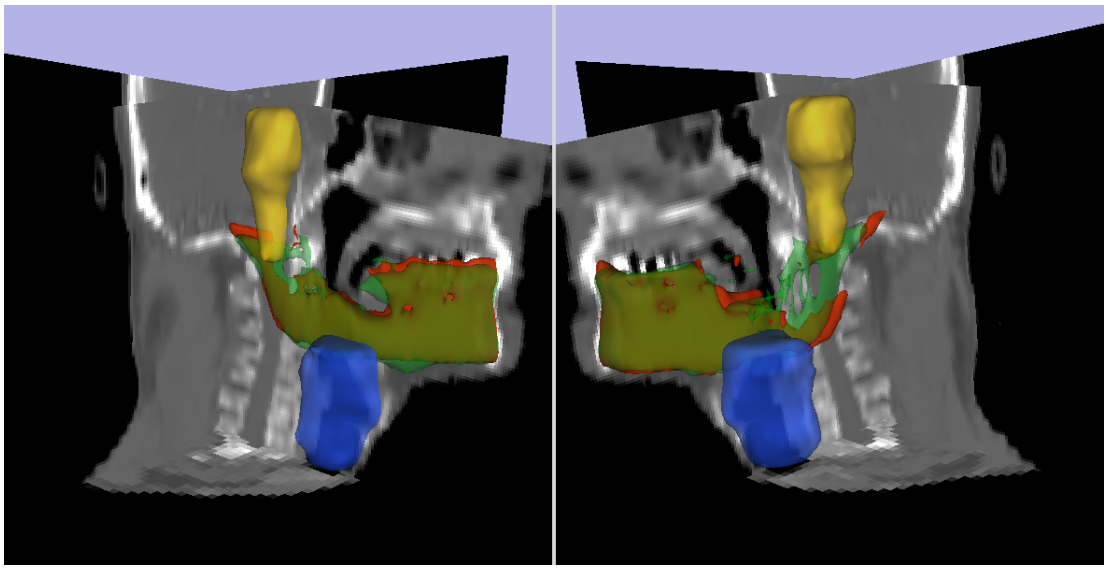


Figure 16: This figure shows the axial, sagittal, and coronal slices from a CT volume. Also, two segmentations, the larynx in blue and the brain stem in yellow, which are assumed to be known, are shown; the ground truth for the mandible's segmentation is in green. Two views of the 3D model are given. The predicted mandible, show in in red, is the shape prior computed by the approach proposed in this paper to be later used by a segmentation algorithm. It is not the result of segmentation.

for a sample $y = (y_1, ..., y_D)^T$ from the multivariate normal distribution with a mean $\mu = (\mu_1, ..., \mu_D)^T$ and covariance matrix Σ . It measures the dissimilarity of the sample from the mean. For computation purposes, we project the observation x onto the latent space using Eq. (55) . Then, the Mahalanobis distance for z^* takes the simplified form $d = D(z^*) = \|z^*\|$. Finally, the confidence metrics in Eq. (45) are defined as

$$p_i = \frac{(\frac{1}{d_i})}{\sum_{j=1}^{k-1}(\frac{1}{d_j})} \ . \tag{58}$$

For this application, the multivariate samples are the segmentations s_k.

3.2.2 Computing the Prediction of the Current Segmentation

We consider the segmentation of each structure s_k as a $P-$dimensional random vector. As alluded to above, PCCA is used to find a pair of linear transformations, one for a random vector $x_1 \in \mathbb{R}^{d_1}$ and the other for random vector $x_2 \in \mathbb{R}^{d_2}$ such that the resulting

projections have one component in the first image correlated to just one component in the second image. In this work, $d_1 = d_2$ because the segmentations are performed on the same image volume and the number of voxel does not change. At the k^{th} stage of the segmentation,a pre-requisite is to have offline PCCA models for these pairs of random vectors: $\{s_k, s_{k-1}\}, ..., \{s_k, s_1\}$. Again, a latent variable model is assumed by PCCA, as in Fig. 17, where z is the latent variable relating the two observed variables x_1 an x_2.

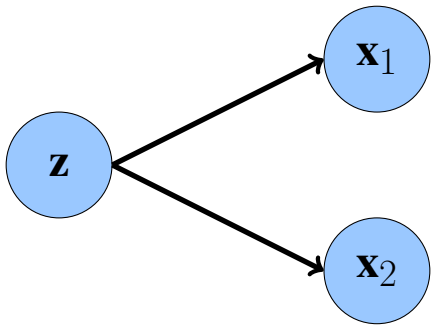


Figure 17: Generative model for PCCA.

In [7], models for z , $z|x_1$, and $z|x_2$ are assumed and Eq. (62) - Eq. (72) are derived using maximum likelihood estimation:

$$z \sim \mathcal{N}(0, I) \tag{59}$$

$$x_1|z \sim \mathcal{N}(W_1 z + \mu_1, \Psi_1) \tag{60}$$

$$x_2|z \sim \mathcal{N}(W_2 z + \mu_2, \Psi_2) \tag{61}$$

$$W_1 = \hat{\Sigma}_{11} U_1 M \tag{62}$$

$$W_2 = \hat{\Sigma}_{22} U_2 M \tag{63}$$

$$M = \Lambda^{\frac{1}{2}} R \tag{64}$$

$$\mu_1 = \hat{\mu}_1 \tag{65}$$

$$\mu_2 = \hat{\mu}_2 \tag{66}$$

$$\mathbb{E}\left[z|x_1\right] = M^T U_1^T (x_1 - \mu_1) \tag{67}$$

$$\mathbb{E}\left[z|x_2\right] = M^T U_2^T (x_2 - \mu_2) \tag{68}$$

$$U_1 = \hat{\Sigma}_{11}^{-\frac{1}{2}} U \tag{69}$$

$$U_2 = \hat{\Sigma}_{22}^{-\frac{1}{2}} V \tag{70}$$

$$K = \hat{\Sigma}_{11}^{-\frac{1}{2}} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-\frac{1}{2}} \tag{71}$$

$$K = U \Lambda V^T \quad . \tag{72}$$

PCCA is carried out by first computing the sample means $\hat{\mu}_1,\ \hat{\mu}_2$ and covariance matrices $\hat{\Sigma}_{11},\ \hat{\Sigma}_{12},$ and $\hat{\Sigma}_{22}$. Then, the matrix K in Eq. (71) is computed and decomposed using singular value decomposition (SVD) into an orthogonal matrix U, a diagonal matrix Λ with positive, decreasing values on the diagonal, and another orthogonal matrix V as in Eq. (72). Using Eq. (69) and Eq. (70), the canonical directions are computed; they are columns of U_1 and U_2 that determine the sub-spaces onto which x_1 and x_2 will be projected, according to Eq. (67) and Eq. (68). Although R can be any orthogonal matrix, it is set to be the identity. A note: the subspaces onto which x_1 and x_2 are projected when computing Eq. (67) and Eq. (68) are the same as the sub-spaces for standard CCA, which is also proved in [7].

Given a previously segmented structure s_j, a vector in $\mathbb{R}^P$, we show how to compute the most likely segmentation for the current structure $s_{k|j}$. In this case, the random vectors x_1 and x_2 will be s_j and s_k, respectively. The most likely latent variable for s_j are called z^{opt}; since the conditional distribution $z|x_1$ is Gaussian,

$$z^{opt} = \mathbb{E}\left[z|s_j\right] \quad . \tag{73}$$

Furthermore, the random variables s_j and s_k conditioned on the latent variable z are independent:

$$f(s_k, s_j|z) = f(s_k|z) f(s_j|z) \quad . \tag{74}$$

The best prediction of s_k given s_j is

$$s_{k|j} = \left[\underset{s_k}{\operatorname{argmax}}\, G(s_k) \right]_{|z^{opt}} \tag{75}$$

where $G(s_k)$ is the probability density function

$$\begin{aligned} G(s_k) = f(s_k|s_j) &= \frac{f(s_k, s_j)}{f(s_j)} \\ &= \frac{\int_\Omega f(s_k, s_j|z) f(z) dz}{f(s_j)} \\ &= \frac{1}{f(s_j)} \int_\Omega f(s_k|z) f(s_j|z) f(z) dz \; . \end{aligned} \tag{76}$$

To maximize $G(s_k)$, the derivative is computed

$$\begin{aligned} \frac{\partial G(s_k)}{\partial s_k} &= \frac{1}{f(s_j)} \int_\Omega \frac{\partial}{\partial s_k} \Big(f(s_k|z) f(s_j|z) f(z) \Big) dz \\ &= \frac{1}{f(s_j)} \int_\Omega \frac{\partial}{\partial s_k} \Big(f(s_k|z) \Big) f(s_j|z) f(z) dz \; , \end{aligned} \tag{77}$$

and since the conditional distribution of s_k is

$$f(s_k|z) = \frac{1}{(2\pi)^{\frac{p}{2}} \left(|\Sigma|\right)^{\frac{1}{2}}} e^{-\frac{1}{2}(s_k - \mu_{s_k|z})^T \Psi^{-1} (s_k - \mu_{s_k|z})} \; , \tag{78}$$

then

$$\frac{\partial}{\partial s_k} \Big(f(s_k|z) \Big) = -\Psi^{-1}(s_k - \mu_{s_k|z}) f(s_k|z).$$

Solving $\frac{\partial G(s_k)}{\partial s_k} = 0$, we find that

$$s_{k|j} = \mu_{s_k|z} = W_2 z^{opt} + \mu_2 \; . \tag{79}$$

3.3 Results

In this section, results are demonstrated that indicate the validity and value of the proposed approach. The data for this experiment was generated synthetically using a single patient scan and the associated segmentations of structures, made by a physician, as the template. The template consists of a segmented larynx, mandible, and brain stem. From this template, a training set of 499 samples for the PPCA and PCCA algorithms was created by applying

a random affine transformation to the template for each sample. The affine transformation allowed for change in scale up to 10 % (increase or decrease) and a translation of up to 6 voxels along each to the x,y,z axes independently. Another sample, number 500, was generated and used for testing. The results of the test are shown in Fig. 16.

It is assumed that the segmentations of two structures, the larynx, the blue structure in Fig. 16, and the brainstem, the yellow structure are available; in practice, the algorithm proposed here is run iteratively, and the two segmentations are simply results from previous stages. So, using s_1 and s_2, we compute $\hat{s}_3$, the segmentation of the mandible, according to Eq. (45). This prediction is shown in red, along with the ground truth in green. It is important to reiterate that $\hat{s}_3$ is not the result of segmentation; it is only a shape prior for segmentation.

3.4 Chapter Conclusion

Canonical correlation analysis is a classical statistical method for measuring relationships between two sets of variables. In this chapter, we describe an approach to use PCCA for predicting the second set of variables in a pair given the first. This approach is justified by presenting an application and testing on sample data. The described method is intended for segmentation of medical images that are degraded by artifacts or contain a target structure that is difficult to distinguish from surrounding tissue by edge or intensity information alone. Physicians can manually perform the segmentation by using their expert knowledge; in other words, they determine a target area on a medical image by using other organs as landmarks combined with their practical knowledge of typical spatial relationships between structures. The novel approach uses two statistical methods, PPCA and PCCA, to build a model for the qualitative approach used by physicians.

CHAPTER IV

A STOCHASTIC APPROACH FOR DIFFEOMORPHIC POINT SET REGISTRATION WITH LANDMARK CONSTRAINTS

Registration is an ill-posed problem and assumptions about the deformation field must be made or knowledge about the expected deformation must be available to compute a particular registration field. The non-rigid registration problem considered in this chapter is challenging because the correspondences and the deformation are unknown; if the correspondences were known, one could perform landmark-based registration by fitting a smooth function to the data [78]. For example, frequently, point sets undergoing registration are sparse representations of real objects. Physically, these objects cannot experience a deformation that is self-intersecting; thus, the field must be constrained to be diffeomorphic. For instance, this restriction is imposed for medical image registration [21, 98] , but state of the art point set registration algorithms do not incorporate it. Furthermore, prior information may be available (e.g., user input, rigidity constraints [57], etc.) and can provide strong clues to guide the registration algorithm. The proposed framework can easily incorporate these types of constraints. We focus on constrained, fully non-rigid registration and devise an algorithm that avoids local minima and limits the model complexity.

Two key features are necessary for a robust and accurate registration algorithm: proper selection of an energy/cost functional and a sound approach for performing the optimization. The suitability of an energy functional is determined by its smoothness (smoothness reduces the quantity of local minima) and the location of its global minimum (the minimum must correspond to the desired solution). The optimization strategy can be split into several broad categories (heuristic-based, gradient descent, probabilistic methods, and filtering methods). The algorithms can be further subdivided based on the deformation

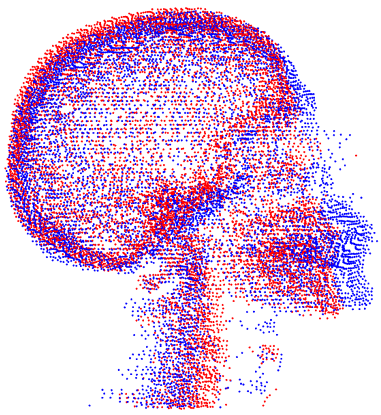


Figure 18: Point clouds generated from two computed tomography (CT) volumes of different patients are shown. The points are sparse representations of the human skeleton; they are generated by applying a threshold to the CT volume to extract a rough bone mask and sampling the mask to obtain the point clouds. Any deformation that proposes to relate them must be injective. The deformations computed with the proposed method are presented in Section 4.5.2.

model (i.e., rigid, affine, piece-wise affine, or non-rigid). Algorithms in this spectra trade off representational flexibility of the deformation for speed and robustness.

This chapter is organized as follows. Section 4.1 reviews related works. Section 4.2 provides the background necessary for Section 4.3. The stochastic point set registration algorithm (SPSR) is presented in Section 4.3, with implementation details provided in Section 4.4. The performance of the algorithm on 2D and 3D data sets is shown in Section 4.5.

Scope In this chapter, we aim to recover a planar transformation that is one to one or a projective transformation that is invertible, called a homography. This class of problems includes medical image registration, e.g. in Fig. 18 , where an injective, physical deformation (i.e., one that does not cause tissue from two different points to map to the same point) is required; also, stereo matching problems, where an object does not self-occlude, also fall in this context.

4.1 Existing Work

Heuristic Methods One of the best known and widely used point set registration algorithms is the iterative closest point (ICP) algorithm [9]. ICP is a popular rigid registration method for performing least squares minimization on the distance between two point sets; as for any gradient descent algorithm, a close initialization of the transformation must be provided to reach a global minimum. In fact, Besl and McKay [9] show that this iterative, heuristic approach to optimization has convergence properties similar to methods using explicit vector gradients, such as steepest descent. In [23], the authors propose a non-rigid robust point matching algorithm. The deformation is parametrized with thin plate splines (TPS) [36] and instead of assigning binary correspondences of data to model points, as ICP does, the authors use soft-assign [73]. Nevertheless, the iterative minimization approach in [23] is quite similar to ICP, but by using fuzzy correspondences and controlling the degree of the fuzzyness with deterministic annealing, the authors obtain a more robust performance especially in the presence of outliers.

Gradient Descent Methods Influence of outliers on the chosen energy functional is a concern for registration; in [94], Tsin and Kanade propose a cost function based on "kernel correlation," analyse its robustness to outliers, and demonstrate results with rigid registration examples. In the same spirit, Jian and Vemuri proposed the Robust Point Set (RPS) registration [10] to compare point sets by forming Gaussian mixtures from the model and data points and computing the integral of the squared difference between the mixtures; this way, each point has limited contribution to the energy.

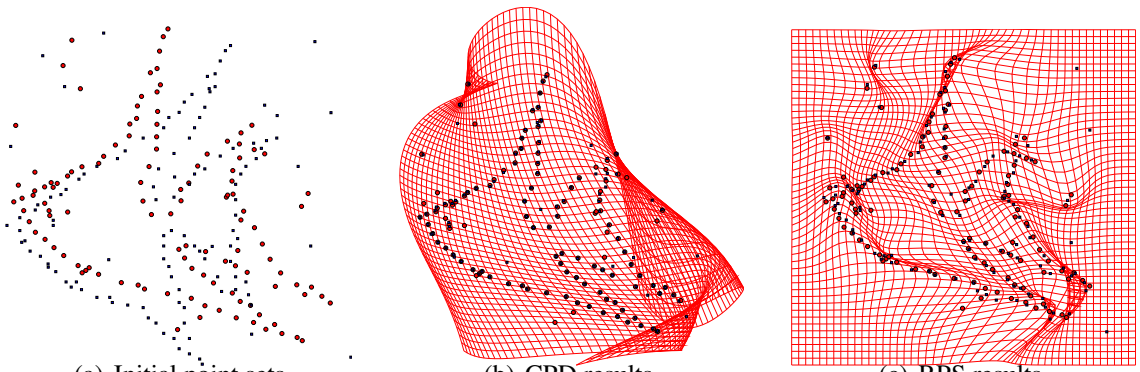
Probabilistic Methods In [22], Chui and Rangarajan obtain the registration field by representing a template point set as a Gaussian mixture model and transforming the data points such that they are maximally explained by the template points. They extend the approach to the non-rigid case and perform optimization through EM. Most recently, Myronenko et

al. [66] introduced the coherent point drift (CPD) algorithm. They replaced the use of TPS in [22] and parametrized the deformation with Gaussian functions instead. The algorithm can be used for data in dimension higher than three where TPS are not defined. Also, [66] removed the simulated annealing step controlling the search range around each model point within the EM algorithm by estimating the parameter instead.

Filtering Methods The filtering approach to registration was introduced in [59] with the unscented particle filter (UPF) applied to rigid registration. The authors use the ICP algorithm to establish correspondences, compute the distance between data sets, and determine the likelihood for the UPF. The Euclidean distance underlying ICP is susceptible to outliers and hence, not always a reliable measure of fit between data sets. In [83], Sandhu *et al.* utilize particle filtering but select an inner product as a similarity measure between the two Gaussian mixtures (one is from the model point set and the other from the data point set). Outliers have local influence leading to a likelihood measure that is more robust. The authors also use a dynamical model to improve convergence speed and robustness. Their method only works for rigid registration. If elastic deformations are allowed, the similarity metric incorrectly moves the data set points toward the mode of the model set's Gaussian mixture density.

Contribution In this chapter, a non-rigid registration framework is described. The approach presented here is not limited to a particular metric or parametrization for the deformation field. For evaluation, an energy for point set registration is selected, the same one used in [10]; however, another cost function can replace it within the same optimization setting, and in fact, even differentiability of the function is not required. The only condition imposed on the energy is that it can be evaluated at any point in the domain.

Our work makes three contributions. First, we demonstrate the need for adding constraints on the registration and show how to enforce them. Two types of constraints are



(a) Initial point sets. (b) CPD results. (c) RPS results.

Figure 19: This figure shows the importance of physically realistic non-rigid deformations; registration results for CPD and RPS(GRBF parametrization) when parameter settings do not provide sufficient regularization can be seen in Fig. 19(b) and 19(c), respectively. Although the points match well, the mapping violates fundamental physical principles; these settings, unknown ahead of time, should be avoided.

presented. The optimal deformation is restricted to be a diffeomorphism; this is not performed by any of the state-of-the art point set registration techniques, to the best of our knowledge. Without the constraint, a computed deformation may not be feasible due to regions with overlap, see Fig. 19 ; hence, it is crucial in any application aiming to register physically meaningful data and makes the registration approach applicable to a wider class of problems. In other scenarios, it may be known that certain image regions can only deform rigidly while others undergo non-rigid deformations or that particular image areas should remain stationary. We show how to take into account this prior knowledge.

Second, we introduce a novel method to regularize the deformation field; instead of a parameter controlling global smoothness as in the RPS and CPD approaches, regularization is implicit through control of the number of basis functions. This difference is vital for accurate registration of data experiencing spatially varying deformations (e.g., Fig. 20); results comparing the two regularization approaches are shown in Section 4.5.1. This problem arises frequently in medical imaging where changes in anatomy are local, requiring a non-smooth deformation field while the remaining domain undergoes smooth, global

changes where regularization is needed to avoid over-fitting. Furthermore, in this formulation, the Gaussian radial basis functions (GRBFs) composing the deformation are not restricted to being centered on data points, and the bases have independent covariance matrices, as opposed to [10]. This formulation allows a small number of basis function to be used in the representation.

Third, we propose simulated annealing for optimizing the cost function and introduce a generating function for the next state relying on particle filters; the merits of this optimization approach can be seen in Section 4.5.1 where, using the same distance metric as RPS, SPSR achieves more accurate registration results because it find a better minimum for the energy. In the RPS and CPD approaches, the optimization is done by gradient descent and EM, respectively; both of these optimization techniques are known to be susceptible to local minima and the registration algorithms inherit their convergence properties.

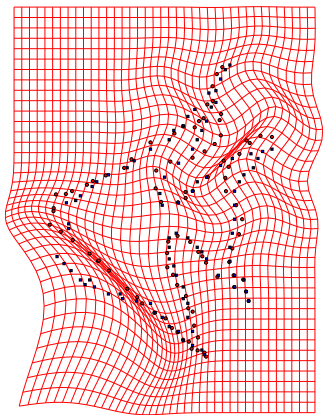


Figure 20: A synthetic example of a deformation field exhibiting spatially varying smoothness. A low frequency perturbation is present in the bottom, left and a high frequency component deform the upper, right portion of the subdomain. While performing the registration, a Tikhonov regularization as used in CPD and RPS, based on the assumption of a globally smooth field, prevents the high frequency components from being recovered.

4.2 Preliminaries

4.2.1 Point Set Registration Problem

Given a point set and a target point set, registration is the task of finding a mapping between the two sets with the appropriate properties. The target point set is called the model M with

$M = \{m_1, ..., m_K\}$, and the given point set is called the data D with $D = \{d_1, ..., d_L\}$. The points $m_i, d_j \in \mathbb{R}^l$ where $l = 2$ or 3 for our experiments. The objective is to find a mapping $\mathcal{L} : \mathbb{R}^l \to \mathbb{R}^l$ that minimizes the distance, d, between the sets M and D :

$$\min_{\mathcal{L}} d(M, \mathcal{L}(D)). \tag{80}$$

One approach to finding $\mathcal{L}$ is to first determine correspondences then to fit a deformation $\mathcal{L}$ to the correspondences. We attempt to solve this problem by finding the deformation directly. The direct approach is preferred because the number of points in each point set may not be equal (correspondences are not one-to-one) or the point sets may have noise, which means a point may not correspond exactly to any point of the other set.

4.2.2 Defining a distance metric

The distance d will depend on the representation chosen for the point sets. Here, we represent the data and model sets as kernel density estimates (KDEs), as in [43]. Identical, symmetric Gaussian kernels are placed around each point in the set to define the Gaussian mixture model for each point set,

$$m(\vec{x}; M) = \frac{1}{K} \sum_{i=1}^{K} \mathcal{N}\big(\vec{x} \,|\, m_i, \sigma^2 I\big), \tag{81}$$

$$c(\vec{x}; \mathcal{L}(D)) = \frac{1}{L} \sum_{j=1}^{L} \mathcal{N}\big(\vec{x} \,|\, \mathcal{L}(d_j), \sigma^2 I\big). \tag{82}$$

The point sets are maximally aligned when their corresponding kernel density estimates are maximally similar. A variety of information theoretic metrics exist to quantify the similarity between densities [10]. Here, the L_2 or integrated square error (ISE) is used,

$$d(M, \mathcal{L}(D)) = \int_{\Omega} \left(m(\vec{x}; M) - c(\vec{x}; \mathcal{L}(D))\right)^2 d\Omega, \tag{83}$$

because a closed form expression for the distance can be computed from the identity

$$\int_{\Omega} \mathcal{N}(\vec{x} \,|\, \vec{\mu}_1, \Sigma_1) \mathcal{N}(\vec{x} \,|\, \vec{\mu}_2, \Sigma_2) d\Omega = \mathcal{N}(\vec{\mu}_1 \,|\, \vec{\mu}_2, \Sigma_1 + \Sigma_2) \; . \tag{84}$$

4.2.3 Parametrizing the Displacement Function

The space of non-rigid transformations is infinite-dimensional, which complicates the optimization of Eq. (80). To improve the tractability of the optimization, we limit the degrees of freedom of the deformation field $\mathcal{L}$ by defining the transformation to be an additive composition of a rigid transformation and a collection of non-rigid basis transformations through a GRBF network,

$$\mathcal{L}(\vec{x};\Theta) = \mathcal{L}_{lin} + \mathcal{L}_{nl} = A\vec{x} + \vec{b} + \sum_{i=1}^{N} \vec{w}_i \mathcal{N}\left(\vec{x} \,|\, \vec{\mu}_i, \sigma_i^2 I\right) . \tag{85}$$

The optimization in Eq. (80) is now over the rigid transformation parameters and the GRBF parameters. Finding the optimal $\mathcal{L}$ is equivalent to determining the parameters Θ:

$$\begin{aligned}
\Theta &= [\Theta_r, \Theta_g] \quad \text{where} \\
\Theta_r &= \left[A_p, \vec{b}\right] \\
\Theta_g &= [\vec{\mu}_1, \vec{w}_1, \sigma_1, \cdots, \vec{\mu}_N, \vec{w}_N, \sigma_N] \\
A_p &= [w, x, y, z], \\
\vec{b} &= [a, b, c], \\
\vec{\mu}_i &= [\mu_{x_i}, \mu_{y_i}, \mu_{z_i}], \text{ and} \\
\vec{w}_i &= [w_{x_i}, w_{y_i}, w_{z_i}] .
\end{aligned} \tag{86}$$

for the 3D case (the 2D case is simpler). The parameters A_p are quaternion coordinates that generate the scaled, rigid transformation matrix A, where s is the scale and $[\alpha, \psi, \phi]$ are the rotation angles. By representing $\mathcal{L}(\cdot)$ as in Eq. (85), the problem from Eq. (80) becomes one of parameter estimation:

$$\min_{\Theta} \; E(\Theta) \text{ with } E(\Theta) = d(M, \mathcal{L}(D;\Theta)) . \tag{87}$$

The representational capacity of the deformation by a GRBF network is guaranteed by the *universal approximation theorem* for radial basis function networks.

Theorem: [70] Let the family $S_{\mathcal{K}}$ consist of functions $q : \mathcal{R}^r \to \mathcal{R}$, $q(x) = \sum_{i=1}^{N} w_i \cdot \mathcal{K}(\frac{x-t_i}{\sigma}),\ N \in \mathcal{N},\ \sigma > 0,\ w_i \in \mathcal{R},\ t_i \in \mathcal{R}^r$. Let $\mathcal{K} : \mathcal{R}^r \to \mathcal{R}$ be an integrable bounded

function such that $\mathcal{K}$ is continuous almost everywhere and $\int_{\mathcal{R}^r} \mathcal{K}(x)dx \neq 0$. Then family $S_{\mathcal{K}}$ is dense in $L^p(\mathcal{R}^r)$ for every $p \in [0, \infty)$.

The symmetric Gaussian function satisfies the properties required of $\mathcal{K}$. Relaxing the constraint of equal σ, it is still possible to find an N such that $\|f - \mathcal{L}\|_{L_2} < \epsilon$, for a given ϵ. Although with a sufficiently high N the linear term in Eq. (85) is unnecessary, it is included to reduce the number of GRBFs needed; the rigid transformation terms account for global movement and the GRBF network represents local deformations. While a large N would lead to increasingly accurate non-rigid registration, we propose to use a small to moderate value of N (10 to 20) to limit variation in the transformation and prevent fitting to noise or other sources of error.

4.2.4 Sequential Monte Carlo

Sequential Monte Carlo samplers [26, 27] draw samples from a sequence of probability measures, each admitting a probability distribution $\{\pi_k(x_{1:k})\}_{k\in\mathbb{T}}$ where $\mathbb{T} = \{1, ..., b\}$ and $x_{1:k} \triangleq (x_1, ..., x_k)$. Then, the probability density function (pdf) can be written as

$$\pi_n(x_{1:k}) = \frac{\gamma_k(x_{1:k})}{Z_k} \tag{88}$$

where Z_k is the normalizing constant

$$Z_k = \int \gamma_k(x_{1:k})\, dx_{1:k}\ . \tag{89}$$

At each instance of time, kept track by the counter variable k, the dimension of the space E_k over which $\pi_k(x_{1:k})$ is defined increases such that $dim(E_{k-1}) < dim(E_k)$, and of course, the dimension of the sample $X_{1:k} \sim \pi_k(x_{1:k})$ also increases with k.

Although it is desired to sample from the pdf $\pi_k(x_{1:k})$, in most cases, sampling directly from $\pi_k(x_{1:k})$ is not feasible due to the complicated nature of this distribution. Rather, importance sampling is used: for a proposal distribution $q_k(x_{1:k})$, the relation

$$\pi_k(x_{1:k}) \propto q_k(x_{1:k}) \cdot \frac{\gamma_k(x_{1:k})}{q_k(x_{1:k})} \tag{90}$$

holds. A proposal distribution is chosen such that $X^i_{1:k}$ for $i \in [1, ..., N]$, N samples at time k, can be drawn. Then, an empirical estimate of the distribution $\pi_k(x_{1:k})$ is expressed as a weighted sum of these particles:

$$\hat{\pi}_n(x_{1:k}) = \sum_{i=1}^{N} W^i_k \delta_{X^i_{1:k}}(x_{1:k}) \; . \tag{91}$$

From Eq. (90), it is clear that the weights are equal to

$$w_k(x_{1:k}) = \frac{\gamma_k(x_{1:k})}{q_k(x_{1:k})} \tag{92}$$

and the normalized weights appearing in Eq. (91) are computed as

$$W^i_k = \frac{w_k(X^i_{1:k})}{\sum_{j=1}^{N} w_k(X^j_{1:k})} \quad . \tag{93}$$

The underlying assumption in Eq. (90) is that a proposal distribution that is easily sampled can be found at each time step. The proposal distribution q_k should have the same support as the original pdf π_k and a similar shape. For an arbitrary $\pi_k(x_{1:k})$, choosing such a function is not trivial. Ideally, the pdfs π_{k-1} and π_k at consecutive time points are not too different, and there is a way to move particles $X^i_{1:k-1}$ to high probability areas of $\pi_k(x_{1:k})$. A solution is to build the proposal distribution iteratively as

$$q_k(x_{1:k}) = \int_{x_{1:k-1}} q_{k-1}(x_{1:k-1}) K_k(x_{1:k-1}, x_{1:k}) dx_{1:k-1} \tag{94}$$

where $K_k(\cdot, \cdot)$ is a transition kernel $K_k : E_{k-1} \rightarrow E_k$. Thus, starting from an initial importance distribution $q_0(x_0)$, a q_k is available at all future time points by forming a series of proposal distributions using Eq. (94). We consider two applications of SMC sampler: for Bayesian filtering in Section 4.2.5 and for optimization in Section 4.2.7.

4.2.5 Particle Filtering

Sequential Bayesian filtering estimation with Monte Carlo simulation, called *particle filtering*, was first introduced by Gordon [34]. In recent years, it has proven to be a powerful scheme for non-linear and non-Gaussian estimation problems due to its simplicity and versatility.

It is common to receive measurements at discrete time points k, and the discrete-time filtering problem is formulated as follows. Let x_k be an unobservable state and $x_{1:k}$ be the history of states up to time k; similarly, let y_k be observable measurements and $y_{1:k}$ be the history of these measurements. Then, the transition equation Eq. (95) and the measurement equation Eq. (96) for the general Markov state-space model are

$$x_k = f_k(x_{k-1}, u_{k-1}) \,, \tag{95}$$

$$y_k = h_k(x_k, v_k) \tag{96}$$

and the conditional distributions are

$$X_k|(X_{k-1} = x_{k-1}) \sim p(x_k|x_{k-1}) \tag{97}$$

$$Y_k|(X_k = x_k) \sim p(y_k|x_k) \,. \tag{98}$$

Here, f_k, h_k are (potentially) time-varying non-linear functions and u_k, v_k are independent and identically distributed (iid) random variables representing noise in the state and measurement equations, respectively, with known probability density functions. Thus, Eq. (95) and Eq. (96) implicitly define the state transition and measurement probabilities $p(x_k|x_{k-1})$ and $p(y_k|x_k)$, respectively.

Bayesian filtering is a natural application of the SMC sampler described in Section 4.2.4. The objective of Bayesian filtering is to estimate a sequence of probability density functions $\{\, p(x_{1:k}|y_{1:k}) \,\}_{k\geq 1}$. The link to the SMC sampler becomes clear if, from Eq. (88), we define $\pi_k(x_{1:k}) = p(x_{1:k}|y_{1:k})$, $\gamma_k(x_{1:k}) = p(x_{1:k}, y_{1:k})$, and $Z_k = p(y_{1:k})$.

Using the conditional distribution of the state variables given the sequence of measurements, an estimate of the current latent state can be computed from some statistic of the pdf (e.g., maximum *a posteriori* estimate). Consequently, obtaining an approximation of the posterior distributions is the subject of the following discussion. If drawing N independent samples $\{X^i_{1:k}\}_{i=1,\ldots,N}$ according to $X^i_{1:k} \sim p(x_{1:k}|y_{1:k})$ were possible, as $N \to \infty$, an

arbitrarily accurate empirical estimate of the posterior pdf could be constructed as

$$p(x_{1:k}|y_{1:k}) \approx \sum_{i=1}^{N} \delta(x_{1:k} - X_{1:k}^i) \ , \tag{99}$$

where δ denotes the Dirac delta function. However, generating samples from the posterior distribution $p(x_{1:k}|y_{1:k})$ is usually not possible (e.g., f and g are non-linear); instead, importance sampling is used.

To fix the computational complexity of updating the proposal distribution, the following form for $q_k(x_{1:k})$ is assumed

$$q_k(x_{1:k}) = q_1(x_1) \prod_{n=2}^{k} q_n(x_n|x_{1:n-1}) \ . \tag{100}$$

The optimal choice for $q_k(x_k|x_{1:k-1})$ takes the form

$$q_k(x_k|x_{1:k-1}) = q(x_k|y_k, x_{k-1}) \tag{101}$$

[30]. In general, the support of $q_k(x_{1:k})$ must coincide with the support of $p(x_{1:k}|y_{1:k})$ and there should be a procedure for drawing samples from this proposal distribution directly. Ideally, the proposal distribution should approximate $p(x_{1:k}|y_{1:k})$.

Since samples are drawn from $X_k^i \sim q(x_k|y_k, x_{k-1})$, they must be weighted, and the approximation of the posterior distribution in Eq. (99) becomes

$$p(x_{1:k}|y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{1:k} - X_{1:k}^i) \ . \tag{102}$$

This can be seen by using the Markov chain assumptions in Eq. (95)-(96) to establish the recursion

$$p(x_{1:k}, y_{1:k}) = p(x_{1:k-1}, y_{1:k-1}) p(x_k|x_{k-1}) p(y_k|x_k) \ , \tag{103}$$

Then, the recursion for the posterior distributions is

$$p(x_{1:k}|y_{1:k}) = p(x_{1:k-1}|y_{1:k-1}) \frac{p(x_k|x_{k-1}) p(y_k|x_k)}{p(y_k|y_{1:k-1})} \ \text{ and} \tag{104}$$

$$p(x_{1:k}|y_{1:k}) \propto \left[p(x_{1:k-1}|y_{1:k-1}) \frac{p(x_k|x_{k-1}) p(y_k|x_k)}{q(x_k|x_{k-1}, y_k)} \right] \cdot q(x_k|x_{k-1}, y_k) \ . \tag{105}$$

Thus, the importance weights in Eq. (102) are update according to the following equation:

$$w_k^i \propto w_{k-1}^i \, \frac{p(x_k^i \mid x_{k-1}^i) p(y_k \mid x_k^i)}{q(x_k^i \mid x_{k-1}^i, y_k)} \; . \tag{106}$$

The generic algorithm begins by first sampling N times from initial state distribution, $p(x_0)$. Following this, the algorithm can be decomposed into two steps: the prediction step and the update step. Using importance sampling [29], the **prediction step** is the act of drawing N samples from the alternative proposal distribution $q(x_k|x_{k-1}, y_k)$. As new information arrives on-line at time k in the form of observation y_k, it is necessary to evaluate the "fitness" of the samples given the new measurement. In other words, as y_k becomes available, the measurement or **update step** in particle filtering is incorporated through the importance weights by equation Eq. (106).

The quantity $p(y_k \mid x_k^i)$ is the unnormalized likelihood of receiving a new measurement y_k given that the state at time k is x_k^i. Also, w_k^i is the normalized weight of i-th particle such that $\sum_{i=1}^N w_k^i = 1$. If a transitional prior $q(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1})$ is employed and re-sampling is performed at each time step so that $w_{k-1}^i = \frac{1}{N}$, the weight are computed simply as

$$w_k^i \propto p(y_k|x_k^i) \quad . \tag{107}$$

Importance sampling causes most particles to have negligible weight after a few iterations, which is called the *sampling degeneracy* problem. To avoid this issue, one can apply a re-sampling scheme, which is generally done by replicating particles in proportion to their weights. This process eliminates samples with low weights and chooses better particles [76]. On the other hand, it produces the loss of diversity for a set of particles, *i.e.*, particles with high weights are selected too often, and thus, the others disappear with time. Therefore, all of the particles will eventually collapse to the same value. To alleviate this *sample impoverishment*, techniques have been proposed to improve the sample diversity such as the regularized particle filter [65] and the Markov chain Monte Carlo (MCMC) move step [33]; see [76] for a detailed discussion.

The computational complexity of the particle filter depends heavily on the dimension of the state vector because particle filters require an increasingly large number of particles for higher-dimensional systems. Choosing a good importance density helps in the reduction of computational complexity [35] but is difficult in practice.

4.2.6 Particle Filtering for Global Optimization

The static problem of finding the global minimum $x^* = \underset{x}{\operatorname{argmin}}\ R(x)$ through stochastic optimization can be phrased in a principled manner as a bayesian filtering problem discussed in Section 4.2.5. One formulation is presented in [107]; below, we present a different formulation. To this end, the following state-space model is defined:

$$x_{k+1} = d(x_k)\ + u_k \tag{108}$$

$$y_k \quad = R(x_k) + v_k \tag{109}$$

where u_k, v_k are i.i.d. random variables $u_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, 1)$. These variables represent the uncertainty in the state update and the state measurement; they implicitly define the probabilities

$$p(x_k|x_{k-1}) = \frac{1}{\left((2\pi)^d\,|Q|\right)^{\frac{1}{2}}} e^{-\frac{1}{2}(x_k - d(x_{k-1}))^T Q^{-1}(x_k - d(x_{k-1}))} \tag{110}$$

$$p(y_k|x_k) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y_k - R(x_k))^2}\ . \tag{111}$$

Here, the history of states is not important and the aim is to estimate the marginalized version of $p(x_{1:k}|y_{1:k})$, $p(x_k|y_{1:k})$, which can be split into two steps [20]:

Predict: Chapman-Kolmogorov Equation

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{112}$$

Update: Bayes Theorem

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}\ . \tag{113}$$

Thus, the equation incorporating a new measurement y_k is

$$p(x_k|y_{1:k}) \propto \int p(y_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \ . \tag{114}$$

A transitional prior, $q(x_k|x_{k-1}, y_k) = p(x_k|x_{k-1})$ is assumed and this process is implemented as follows:

Algorithm 1 An implementation of the particle filter.

initialize

$\left\{\tilde{X}_0^i\right\} \sim q_0(x) = p(x_0)$ ▷ Draw N iid samples

$w_0(\tilde{X}_0^i) = \frac{1}{N}$

procedure PARTICLE FILTER

for $k = 1:$ maxIterations **do**

Sample: $\left\{X_k^i\right\} \sim p(x_k|\tilde{X}_{k-1}^i)$

Importance Weights:

$W_k(X_k^i) = \frac{1}{N} \cdot \alpha_k(X_k^i)$

where $\alpha_k(X_k^i) = p(y_k|X_k^i)$

Normalize Weights : $w_k^i = \frac{W_k(X_k^i)}{\sum_{i=1}^N W_k(X_k^i)}$

Approximate : $p(x_k|y_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - X_k^i)$

Re-sample: $\left\{\tilde{X}_k^i\right\} \sim p(x_k|y_{1:k})$, $w_k(\tilde{X}_k^i) = \frac{1}{N}$ ▷ obtain N iid samples

Translating the static optimization problem to the filtering context requires a definition for the system being observed through measurements y_k. It is assumed that the realizations observed are equal to the optimal value of $R(x)$, i.e., $y_k = R(x^*)\ \forall k \in \{0, 1, ...\}$. Of course, x^* is unknown *a priori* as is $R(x^*)$. Instead, $y_k = g$ is used where g is a lower bound for $R(x)$:

$$g \leq R(x) \ , \ \forall x \in S \quad \text{where } S = \text{ domain of } R(x) \ . \tag{115}$$

From Eq. (111), if $R(x_k^i) < R(x_k^j)$ then $p(y_k|x_k^i) > p(y_k|x_k^j)$. Consequently, from Algorithm 1 it is clear that as $k \to \infty$, the filter converges to

$$p(x_k|y_{1:k}) = \delta(x_k - x^*) \ . \tag{116}$$

In the case of Eq. (87), an obvious choice for g is $g = 0$ since the integral L_2 distance is always positive. A trivial choice for the state dynamics, $d(\cdot)$, is $d(x_k) = x_k$, which corresponds to a random walker, according to Eq. (108). More sophisticated local exploration approaches can be used (e.g., [42]) to explore the state space efficiently.

4.2.7 Simulated Annealing

Simulated annealing is a Markov Chain Monte Carlo (MCMC) method for global optimization [50] that samples the sequence of probability distributions $\{p_k(x)\}_{k\in\mathbb{T}}$ with $\mathbb{T} = \{1, ..., b\}$,

$$p_k(x) \propto [p(x)]^{\frac{1}{T_k}} \quad \text{where } 0 < T_b < ... < T_1 \text{ and } 1 << \frac{1}{T_b} \, . \tag{117}$$

Suppose it is desired to obtain the global maximum of a distribution $p(x)$. One inefficient solution is to sample from $p(x)$ using a markov chain monte carlo (MCMC) approach to produce samples $X^i \sim p(x)$ for $i \in [1, ..., N]$ and choose the maximum according to

$$X^* = \underset{X_i}{\operatorname{argmax}}\, p(X^i) \, . \tag{118}$$

Unless $p(x)$ has significant probability mass near the optimum, the sampler will likely spend much of the computation time in regions distant from the global mode. Instead, an annealing approach is can be used: samples are taken from the sequence of pdfs in Eq. (117) to encourage sampling from around the global maxima of $p(x)$ [26] as $T_b \to 0$.

Since $p_k(x)$ is usually not one of the easy to sample, standard distributions, sequential importance sampling can be applied. The proposal distribution is formed recursively at each time k according to

$$q_k(x_k) = \int_{x_{k-1}} q_{k-1}(x_{k-1}) K_k(x_{k-1}, x_k) \, dx_{k-1} \tag{119}$$

which is clearly the marginalized version of Eq. (94) and again defined by the Markov transition kernel $K_k(x_{k-1}, x_k)$. Rather than attempting to compute this (typically intractable)

integral, we introduce an artificial backwards kernel $L_{k-1} : E \times E \rightarrow [0, 1]$ for each transition kernel $K_k(x_{k-1}, x_k)$ and define an artificial joint distribution

$$\tilde{p}_k(x_{1:k}) = \frac{\tilde{\gamma}(x_{1:k})}{\tilde{Z}_k} \text{ where} \tag{120}$$

$$\tilde{\gamma}_k(x_{1:k}) = \gamma_k(x_k) \prod_{n=1}^{k-1} L_n(x_{n+1}, x_n) \tag{121}$$

whose marginal distribution is $p_k(x_k)$. Once $L_n(x_{n+1}, x_n)$ is selected, phrasing the annealing problem in the SMC framework is complete. The unnormalized weights are [26]

$$w_k(x_{1:k}) = w_{k-1}(x_{1:k-1})\tilde{w}_k(x_{k-1}, x_n) \text{ where} \tag{122}$$

$$\tilde{w}_k(x_{k-1}, x_k) = \frac{\gamma_n(x_k)L_{k-1}(x_k, x_{k-1})}{\gamma_{k-1}(x_{k-1})K_k(x_{k-1}, x_k)} \, . \tag{123}$$

Now, suppose we are interested in finding the global optimum of a multivariate function $R(x)$; a particular choice of R, useful for point set registration, is given in Section 4.3.1 but is abstracted out for the purpose of this discussion. Providing an acceptable initialization for the parameter vector is typically non-trivial and the likelihood of a gradient-based algorithm to become trapped in a local minimum is high. Consequently, we propose a stochastic method based on simulated annealing to find the minimum x^* of the objective function $R(x)$. In Eq. (117), $p(x) \propto e^{-R(x)}$, and as $b \rightarrow \infty$ samples are drawn from around the global maxima of $p(x)$ [26].

Extensions to continuous variables are described in [56, 96]. The steps for finding an optimal value using the general continuous simulated annealing (CSA) algorithm are:

1) Initialize state variable $\Theta_0, s_0 = \{\Theta_0\}, k = 0, t_0 = 1$

2) Generate next candidate state by $\tilde{\Theta}_{k+1} \sim G(\Theta_k, \Theta_{k+1})$

3) Accept/reject candidate state with the following rule:

Generate a random number p from a uniform distribution $\mathcal{U}(0,1)$

$$\theta_k = \begin{cases} \tilde{\Theta}_{k+1} & \text{if } p \leq \mathcal{A}(\Theta_k, \tilde{\Theta}_{k+1}, t_k) \text{ (i.e. accepted)} \\ \Theta_k & \text{otherwise} \qquad\qquad \text{(i.e. rejected)} \end{cases}$$

t_k is temperature of annealing at time k

4) Record traversed states until time k , $s_{k+1} = s_k \cup \{\Theta_k\}$

5) Update temperature according to the cooling schedule U, $t_{k+1} = U(k+1)$.

6) If stopping criteria is not reached, $k = k+1$ and go to Step 1 .

Simulated annealing is a Metropolis-Hastings (M-H) algorithm with a stationary distribution $p_k(x)$, defined in Eq. (117). Hence, the acceptance function is defined as

$$\mathcal{A}(x_k, \tilde{x}_{k+1}, T_k) = \min\left\{1, \frac{p^{\frac{1}{T_i}}(\tilde{x}_{k+1})G(x_k|\tilde{x}_{k+1})}{p^{\frac{1}{T_i}}(x_k)G(\tilde{x}_{k+1}|x_k)}\right\} , \tag{124}$$

and if the transition kernel is reversible (i.e., $G(x_k|\tilde{x}_{k+1}) = G(\tilde{x}_{k+1}|x_k)$), it can be simplified to

$$\mathcal{A}(x_k, \tilde{x}_{k+1}, T_k) = \min\left\{1, \frac{p^{\frac{1}{T_i}}(\tilde{x}_{k+1})}{p^{\frac{1}{T_i}}(x_k)}\right\} . \tag{125}$$

In the M-H formulation [77, 4] , the detailed balance, which leads to an acceptance function requiring a reversible kernel, is stated as follows

$$p(x_k)G(x_{k-1}|x_k) = p(x_{k-1})G(x_k|x_{k-1}) . \tag{126}$$

This condition is sufficient (not necessary) to ensure that the function from which the MCMC is sampling remains the desired invariant distribution $p(x)$ since

$$p(x_k) = \int p(x_{k-1})G(x_k|x_{k-1})dx_{k-1} . \tag{127}$$

However, since the temperature T_k is being changed at each time, Eq. (127) does not hold and maintaining detailed balance by restricting the transition kernel to be reversible is not necessary. We relax this constraint.

Furthermore, simulated annealing can be seen as sampling from a sequence of distributions $p_k(x) = p^{\frac{1}{T_k}}(x)$ [26] using a single particle and the SMC framework of Section 4.2.4. There are no restrictions placed on the transition kernel in Eq. (119). And, since sampling from the sequence $p_k(x)$ exactly is not necessary for finding the optimum, the only desired property is that the chosen kernel asymptotically samples from around global maxima of $p(x)$.

4.3 Constrained, Stochastic Point Set Registration Algorithm

4.3.1 Examples of Useful Constraints

A measure of similarity between two point sets was presented in Section 4.2.1; along with other examples in [10, 23, 66], the corresponding algorithms optimize the unconstrained problems to achieve registration. Registration is an ill-posed problem and the commonly used Tikhonov approach is not the only way or the best way to regularize the problem. This work focuses on meaningful regularization of the deformation field by adding constraints to the existing objective functions (e.g., Eq. (87)). Constraints are useful for enforcing smoothness(e.g., maintaining a positive definite Jacobian) and restricting the allowed deformations. We recognize these needs and the optimization approach proposed in Section 4.3.2 is well suited to include constraints. In this section, two examples of frequently applicable constraints are presented; they are not required to execute SPSR but are often useful in the types of problems that are considered. Also, the constraints are not the only ones possible within the proposed registration framework but are meant to serve as examples.

The first type of constraint is the injectivity constraint. The data points considered in this work, are sparse representations of real objects with physical meaning (e.g., landmarks in a medical image). Hence, the deformation sought is the one that aligns the two sets as well as possible while respecting physical constraints. One such constraint is that two

different points in space cannot map to the same point. When the constraint fails, the deformation field appears to overlap itself, as per Fig. 19. Respecting this non-overlapping constraint requires a one-to-one mapping of the domain under $\mathcal{L}$. A second constraint is that the transformation must preserve the orientation of the space. These constraints are not strictly enforced by any state-of-the-art registration algorithms discussed in Section 4.1. Uniqueness of the mapping and preservation of orientation can be tested through the Jacobian of $\mathcal{L}$ from Eq. (85), with the Jacobian denoted by J. In particular, enforcing the constraint

$$d_J(\vec{x}) = \det(J(\mathcal{L}(\vec{x};\Theta))) > 0 \quad , \forall \vec{x} \in S, \tag{128}$$

will ensure an orientation preserving and one-to-one transformation [61]. The function $\det(\cdot)$ is the determinant, and $S \subset \mathbb{R}^l$ is an open subset containing the region of interest.

Another constraint arises from the necessity to keep certain points/regions in the domain stationary while the remainder are free to move in the direction minimizing the energy function. The local rigidity constraint [57] states that points in a subdomain can undergo only a rigid deformation and points outside can move non-rigidly. This constraint can be reduced to the problem of fixing points in a subdomain to be stationary by: rigidly aligning the point sets according to the constrained regions and performing deformable registration subject to the deformation being the identity within these regions. Explicitly, user constraints that prevent chosen points $\vec{x}_1, ..., \vec{x}_C$ from moving are stated as $\mathcal{L}(\vec{x}_i) = \vec{x}_i \; \forall i \in [1, ..., C]$. Thus, the optimization problem becomes

$$\min_{\Theta} E(\mathcal{L}(\vec{x};\Theta)) \tag{129}$$

$$s.t. \quad \mathcal{L}(\vec{x}_1;\Theta) = \vec{x}_1 \tag{130}$$

$$\vdots$$

$$\mathcal{L}(\vec{x}_C;\Theta) = \vec{x}_C$$

$$\det(J(\mathcal{L}(\vec{x};\Theta)) > 0 \; . \tag{131}$$

To enforce the stationarity constraints, we change the family of functions in which the deformation must lie. This change is made by adding Gaussian basis functions of width σ_C^2 centered at the points $\vec{x}_1, ..., \vec{x}_C$ to the original parametrization of $\mathcal{L}_{nl}$ in Eq. (85):

$$\hat{\mathcal{L}}(\vec{x};\Theta) = A\vec{x} + b + \mathcal{L}_{nl}(\vec{x};\Theta) + \sum_{i=1}^{C} \vec{w}_{C_i} \mathcal{N}\big(\vec{x} \,|\, \vec{x}_i, \sigma_C^2 I\big) \text{ where } \vec{w}_{C_i} \text{ satisfy} \tag{132}$$

$$\begin{aligned} \sum_{i=1}^{C} \vec{w}_{C_i} \mathcal{N}\big(\vec{x}_1 \,|\, \vec{x}_i, \sigma_C^2 I\big) &= -\mathcal{L}_{nl}(\vec{x}_1;\Theta) \\ &\vdots \\ \sum_{i=1}^{C} \vec{w}_{C_i} \mathcal{N}\big(\vec{x}_C \,|\, \vec{x}_i, \sigma_C^2 I\big) &= -\mathcal{L}_{nl}(\vec{x}_C;\Theta). \end{aligned} \tag{133}$$

The weights for these new basis functions $\vec{w}_{C_i}$ depend on the parameters Θ and are computed such that $\hat{\mathcal{L}}(\vec{x}_i;\Theta) = \vec{x}_i$ for $i \in [1, C]$. When $\mathcal{L}$ is equal to the identity, $\vec{w}_{C_i} = \vec{0}$. As $\mathcal{L}$ changes from the identity at the constraint point, the linear system of equations in Eq. (133) is solved for $\vec{w}_{C_i}$.

To enforce some/all of the constraints from Eq. (130)-(131), optimization of a cost function $R(\Theta)$ is performed instead of the unconstrained similarity metric in Eq. (87). Three choices for $R(\Theta)$ are presented in Eq. (134)-(136)

$$R_1(\Theta) = E(\mathcal{L}(D;\Theta)) \tag{134}$$

$$R_2(\Theta) = \begin{cases} E(\mathcal{L}(D;\Theta)) & \text{if } r(\cdot) > 0 \\ B \cdot (\|m(\vec{x};M)^2\|_{L^2}^2 + \|c(\vec{x};\mathcal{L}(D;\Theta))\|_{L^2}^2) & \text{if } r(\cdot) \leq 0 \end{cases} \tag{135}$$

$$R_3(\Theta) = \begin{cases} E\Big(\hat{\mathcal{L}}(D;\Theta)\Big) & \text{if } r(\cdot) > 0 \\ B \cdot (\|m(\vec{x};M)^2\|_{L^2}^2 + \|c(\vec{x};\hat{\mathcal{L}}(D;\Theta))\|_{L^2}^2) & \text{if } r(\cdot) \leq 0 \end{cases} \tag{136}$$

where $r(\Theta) = \det(J(\mathcal{L}(\vec{x};\Theta)))$

and the rationale for these choices is described in the subsequent paragraph.

First, for unconstrained optimization of Eq. (87), $R(\Theta) = R_1(\Theta)$ is chosen. Second, the injectivity constraint only, in Eq. (128), is enforced by selecting $R(\Theta) = R_2(\Theta)$ for

optimization. In SPSR, optimization of the function $R(\Theta)$ is performed using a particle filter, as described in Section 4.2.6. The cost function $R_2(\Theta)$, in Eq. (135), imposes a soft constraint on the injectivity of $\mathcal{L}$. This soft constraint holds exactly as $B \to \infty$: if $r(x_n) \leq 0$, then, $p(y_n|x_n) \approx 0$, which leads to the rejection of candidate particles resulting in non-physical deformation fields by setting their likelihood to zero. Finally, choosing $R(\Theta) = R_3(\Theta)$ ensures that the optimal deformation will be invertible and will satisfy the landmark constraints in Eq. (130) by evaluating the likelihood of Θ using the deformation field $\hat{\mathcal{L}}(\vec{x}; \Theta)$.

4.3.2 Optimization Algorithm

A gradient descent method cannot be employed to minimize the cost functions in Eq. (135)-(136) because they are not differentiable and a satisfactory initialization is unavailable. Instead, a stochastic, hierarchical approach is used to find the parameters in Eq. (86). The similarity transformation parameters (rigid rotation, scale, and translation) Θ_s are computed first. Since the dimensionality is low, the global PF optimzation from Section 4.2.6 is used directly to find

$$\hat{\Theta}_s = \min_{\Theta_s} R(\Theta_s|\Theta_g) \; . \tag{137}$$

Now, we introduce the stochastic point set registration (SPSR) algorithm to perform the following optimization:

$$\hat{\Theta}_g = \min_{\Theta_g} R(\Theta_g|\hat{\Theta}_s) \; . \tag{138}$$

The overall strategy is shown in Fig. 21, whose details follow. It is known that the sufficient number of particles for estimating distributions using a particle filter grows exponentially with the dimensionality of the state space. The parameter vector Θ_g of $R(\cdot)$ is large; there are seven parameters per basis function. However, while parameters belonging to the same basis are highly related, there is a weaker correlation between parameters of different bases. Blindly applying a particle filter to correlated variables does not take into account this prior information about their dependence. Hence, rather than optimizing over the entire state

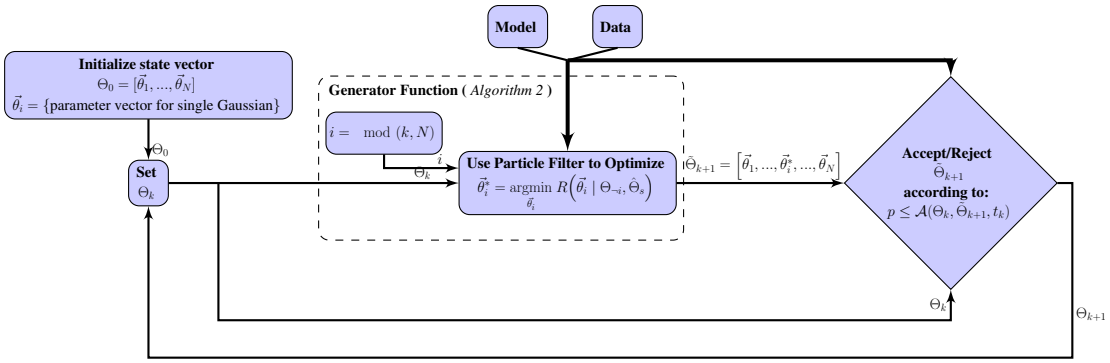


Figure 21: The SPSR registration algorithm described in Section 4.3.2.

vector Θ_g in Eq. (86), we group elements of the state vector into zones as

$$\Theta_g = \left[\vec{\theta}_1, \ldots, \vec{\theta}_N\right] \qquad \text{where} \tag{139}$$
$$\vec{\theta}_i = [\vec{\mu}_i, \vec{w}_i, \sigma_i] \qquad \text{for } i \in [1, N] \; .$$

and minimize $R(\cdot)$ with respect to a single zone $\vec{\theta}_i$ at a time:

$$\vec{\theta}_i^* = \underset{\vec{\theta}_i}{\text{argmin}}\, R\left(\vec{\theta}_i \mid \Theta_{\neg i}, \hat{\Theta}_s\right) \; . \tag{140}$$

The optimization in Eq. (140) is accomplished over the lower dimensional space with the PF, as per Section 4.2.6. In this case, the state x_k being estimated is $x_k = \vec{\theta}_i$.

The optimum with respect to the entire parameter vector Θ_g is achieved using the CSA approach described in Section 4.2.7; this section details the functions $\mathcal{A}(\cdot)$, $G(\cdot, \cdot)$, and $U(\cdot)$. The generator function $G(\tilde{\Theta}_{k+1}, \Theta_k)$ determines how the transition from the current state Θ_k to a proposed state $\tilde{\Theta}_{k+1}$ is made. The particular generator function used for SPSR is defined by Algorithm 2. Here, generating a proposed state involves perturbing a single zone in Eq. (139) while keeping the others constant. In particular, at the k^{th} iteration, this perturbation is made by optimizing the objective function over parameters in zone $i = \text{mod } (k, N)$, as written in Eq. (140), which correspond a single Gaussian basis function.

Algorithm 2 The generator function $G(\tilde{\Theta}_{k+1}, \Theta_k)$ to move between states.

1: Compute $i = \mod(k, N)$,

2: Set $\Theta_{\neg i} = \left[\vec{\theta}_1, ..., \vec{\theta}_{i-1}, \vec{\theta}_{i+1}, ..., \vec{\theta}_N\right]$

3: Use particle filter (Section 4.2.6) to find $\vec{\theta}_i^* = \underset{\vec{\theta}_i}{\operatorname{argmin}} R\Big(\vec{\theta}_i \mid \Theta_{\neg i}, \hat{\Theta}_s\Big)$

4: Set $\tilde{\Theta}_{k+1} = \left[\vec{\theta}_1, ..., \vec{\theta}_{i-1}, \vec{\theta}_i^*, \vec{\theta}_{i+1}, ...\vec{\theta}_N\right]$

The acceptance function used is the Metropolis function

$$\mathcal{A}(\Theta_k, \tilde{\Theta}_{k+1}, t_k) = \min\left\{1, e^{-\frac{R([\hat{\Theta}_s, \tilde{\Theta}_{k+1}]) - R([\hat{\Theta}_s, \Theta_k])}{t_k}}\right\}, \tag{141}$$

and the cooling schedule chosen is

$$U(k) = \frac{R(\left[\hat{\Theta}_s, \Theta_0\right])}{1 + (\lceil \frac{k}{N} \rceil)^2} \tag{142}$$

where the $\lceil \cdot \rceil$ is the ceiling function. The generator function proposed in this section is clearly not reversible, $G(\Theta_{k+1}, \Theta_k) \neq G(\Theta_k, \Theta_{k+1})$. In Metropolis-Hastings (M-H), detailed balance is maintained by construction to ensure sampling from a time-invariant distribution. In simulated annealing, since homogeneity is violated due to the time dependent probability distribution, detailed balance is no longer maintained [4]. Thus, there is no requirement to enforce detailed balance through a symmetric transition kernel [45]. We cannot prove that the exact global minimum will be found with the chosen generator function and cooling schedule, but the proposed algorithm can accept states that cause the cost function $R(\cdot)$ in Eq. (87) to increase and thus, escape local minima. Also, results in Section 4.5 empirically show that better minima are found, since the algorithm outperforms other state of the art registration approaches.

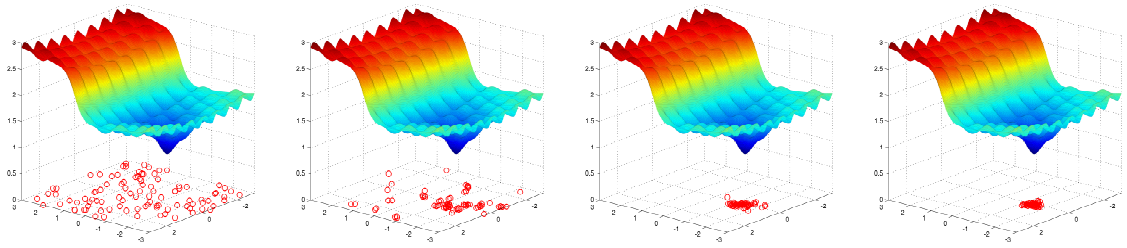
4.3.3 Illustration of the PF Optimization Process

It is expected that $R(\Theta)$ will be a highly non-linear function especially when constraints are added. Since the proposed optimization method is based on the particle filter, in this

section, the PF's ability to locate the global minimum of an energy function with a complicated terrain is illustrated. The following function is used as an example because of its many local minima and a distinct global minimum:

$$g(x,y) = \frac{1}{1+e^{-5(x-\frac{1}{2})}} - e^{-\frac{1}{25}(x^2+y^2)} + \frac{1}{2}e^{-2(x+1.5)^2+y^2} \ldots + \frac{1}{10}\cos\left(\frac{4\pi}{3}x\right)\sin\left(\frac{8\pi}{3}y\right) + 2\,. \tag{143}$$

This function is shown in Fig. 22(a)-22(d) and Fig. 23(a)-23(e) as a surface. For ease of



(a) Initialization. (b) Iteration 4. (c) Iteration 12. (d) Iteration 20.

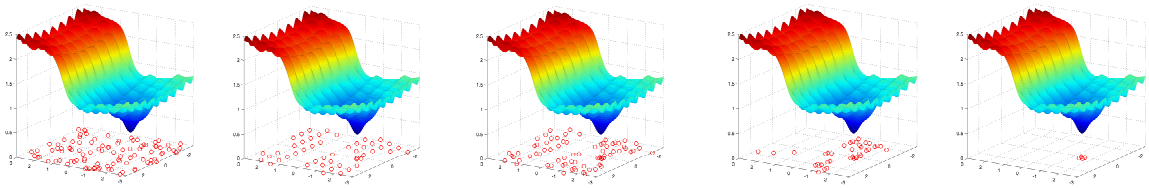
Figure 22: This figure demonstrates optimization of the energy function in Eq. (143) with $d(x_n) = x_n$ in the dynamics equation Eq. (108). The plots include the cost function shown as a surface and the location of 100 particles (red circles) at iterations 0, 4, 12, and 20. It can be seen that by the 20^{th} iteration, all particles are concentrated around the global minimum.

illustration, the state space is two dimensional: the x, y spatial coordinates. Particles in the PF are denoted by red circles and initialized according to $p(x_0)$, which in this case, is a uniform distribution across the state space, as in Fig. 22(a) and Fig. 23(a). This initialization assumes no prior information is available about the location of the optimum; clearly, if some knowledge is obtained *a priori*, the particles can be initialized more efficiently. In this example, 100 particles are used for demonstration although in practice far fewer are sufficient to find the global minimum.

The measurements in Eq. (109) are produced by evaluating the function $g(x,y)$ at the particles' locations. For the same cost function, convergence of the PF using two different function for the dynamics $d(\cdot)$ in Eq. (108) is demonstrated. First, $d(\cdot)$ is chosen as the identity function and the particles propagate according to a noise model: $x_{k+1} = x_k + u_k$.

By the 4^{th} iteration in Fig. 22(b), the majority of the particles are located in the lower half of the function because states which give high values for $g(x, y)$ have a low likelihood, and hence, fewer particles are used to describe that region of the pdf.

Another choice for $d(x_k)$ is to perform a gradient descent operation for a pre-defined number of iterations starting from a given initialization; this operation explores the cost surface locally. The process, using this choice for the dynamics function, to optimize the energy in Eq. (143) is illustrated in Fig. 23. Each iteration of the filter involves two steps. In particular, obtaining the particles' locations at the next time step starting from an initialization x_0, $d(x_0)$ moves each particle to the nearest local minimum as in Fig. 23(b) and the noise model u_0 perturbs them to potentially move the particle to the basin of convergence of a better local minimum, as in Fig. 23(c). The results shown in Fig. 22 should be compared to Fig. 23. In Fig. 23, 100 particles were used; by the 10^{th} iteration, the global minimum has been found. It should be noted that particle filtering in the proposed manner is markedly different from a gradient descent with random restarts: since particles are propagating simultaneously, relative costs are known and computational effort is not spent exploring improbable state regions as seen in Fig. 22(b) and Fig. 23(d).



(a) x_0. (b) $d(x_0)$. (c) $x_1 = d(x_0) + u_0$. (d) x_4. (e) x_{10}.

Figure 23: Optimization process of the energy in Eq. (143) where $d(x_k)$ is a gradient descent operation on $g(x, y)$ starting from $x_k = (x, y)$. Particles are uniformly initialized in Fig. 23(a). State x_1 is computed by performing gradient descent from x_0, $d(x_0)$ (Fig. 23(b)) and perturbing with Gaussian noise u_0 (Fig. 23(c)). In each figure, 100 particles are plotted as red circles, although in some instances multiple particles appear as one due to their virtually identical locations.

Using the gradient descent dynamics, an iteration becomes more computationally expensive, but fewer particles can be used and fewer iteration executed to converge. The

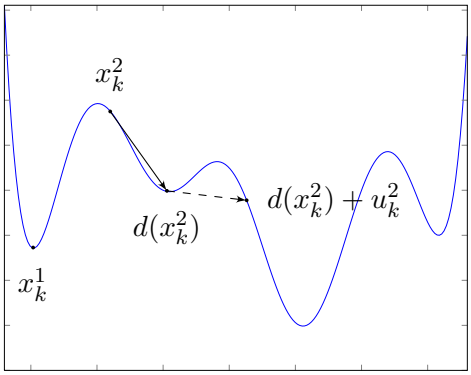


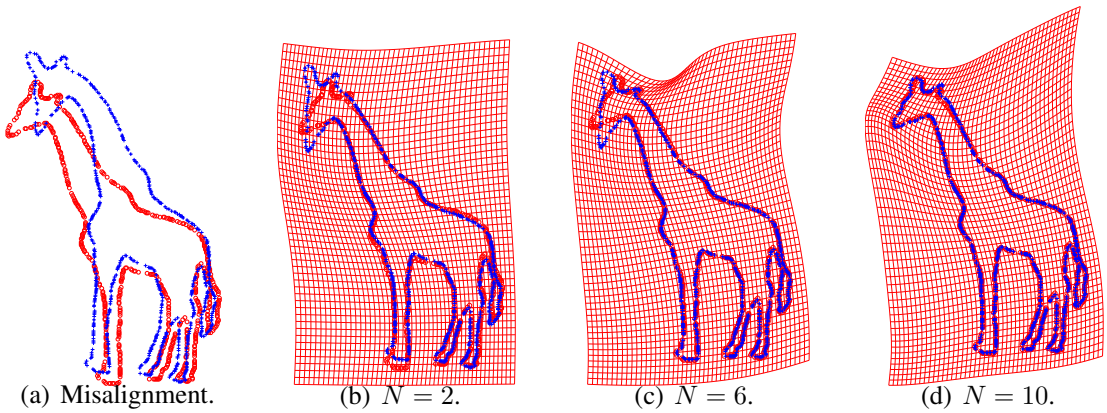
Figure 24: An illustration of each component in the dynamics equation used to propagate a component and efficiently explore the state space. A given particle moves towards a local minimum according to $d(\cdot)$ (e.g., a gradient descent operation) and is perturbed by random noise u_k to potentially move the particle into the basis of convergence for a better local minimum.

reason for this improved convergence can be seen in Fig. 24; note that the subscript refers to the time step while the superscript is the particle index. Consider a single variable cost function $g(x_k)$, plotted in Fig. 24, and two particles from a particle filter x_k^1 and x_k^2. If the value $g(x_k^2)$ was used to determine the probability of state x_k^2, this particle is likely to be discarded due to its high energy value. However, from x_k^2 a low energy state can be reached: the particle moves to $d(x_k^2) + u_k^2$, which is a location that should be considered further.

4.3.4 Illustration of the Registration Process

This section demonstrates the effect of changing the number of basis functions N that are used to represent the deformation field in Eq. (85) and illustrates the registration process presented in Section 4.3.2. An example of two misaligned point sets of a giraffe are shown in Fig. 25(a); misalignment is large around the head of the animal and become increasingly small moving towards the tail. In this experiment, three values of N were used $N = 2, 6, 10$, and a registration using $R_2(\Theta)$ from Eq. (135) was run to convergence for each value of N. The results are shown in Fig. 25(b)-Fig. 25(d), respectively. It is expected that misalignments making the largest contribution to the cost function would be corrected first,

and increasing the degrees of freedom by raising N should allow the algorithm to capture small errors in alignment. Results of this experiment in Fig. 25 support this intuition: with just two basis functions in the deformation, the majority of misalignment is captured. Increasing the number of basis functions to six captures smaller contributions to the energy function by aligning the back and legs of the giraffe. Finally, with ten basis functions, the point set are almost perfectly aligned; small errors are present around the giraffe's ears, which, if desired, can be corrected by increasing N further. For most applications, the results in Fig. 25(d) would be deemed sufficiently close and increasing N unnecessary.



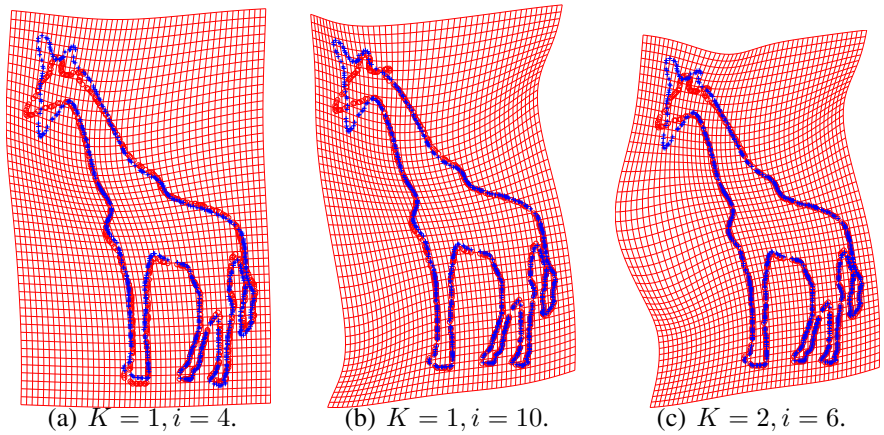
(a) Misalignment. (b) $N = 2$. (c) $N = 6$. (d) $N = 10$.

Figure 25: Illustration of the effect changing the number of basis functions N in Eq. (85) has on registration results. Starting from the original point sets, Fig. 25(a), the final registration for 2, 6, 10 Gaussian basis functions are show in in Fig. 25(b)-25(d), respectively. Registration accuracy is improved as N increases; however, a larger N provides increasingly marginal improvement. Notice the alignment around the giraffe's head, back, legs in each figure.

The sequential nature of the SPSR registration algorithm is illustrated in Fig. 26; this figure should be read simultaneously with Fig. 21. Each subfigure 26(a)-26(f) is the result of generating a state at time k, Θ_k and warping the domain according to the deformation $\mathcal{L}(\vec{x}; \left[\hat{\Theta}_s, \Theta_k\right])$; the transition between states is made according to $G(\tilde{\Theta}_{k+1}, \Theta_k)$ in Algorithm 2. For simplicity, it is assumed that the point sets are rigidly aligned and a purely non-rigid component of the deformation is sought-after.

An optimization "level" K is complete when all of the basis function have been "placed." Initially, the means and weights of all bases are trivially initialized to zero; a basis function

is said to be "placed" at the current "level" after its parameters have been optimized. For $N = 10$, the first level is complete when $i = 10$ since all bases have been placed, and the K^{th} optimization level begins by optimizing over parameters of the first basis, keeping the rest constant according to the values found during level $K - 1$. For example, Fig. 26(a) shows the registration result when parameters for the first four bases have been optimized while the others remain trivially initialized. At $K = 1, i = 10$ in Fig. 26(b), all bases have been optimized. For the second level $K = 2$, Fig. 26(c) shows registration results after parameters for the first six bases have been optimized, one at a time, with respect to the other bases. In this manner, various states continue to be explored until registration is terminated at the end of level three, Fig. 26(f), and alignment is achieved.



(a) $K = 1, i = 4$. (b) $K = 1, i = 10$. (c) $K = 2, i = 6$.

(d) $K = 2, i = 10$. (e) $K = 3, i = 4$. (f) $K = 3, i = 10$.

Figure 26: The registration process described in Fig. 21 is illustrated. The number of basis functions in Eq. (85) is kept constant at $N = 10$. Each subfigure is the result of an instantiation of the state vector Θ_g.

4.4 Implementation Details

4.4.1 Parameter Selection

Implementation of the algorithm requires specifying various parameters associated to the optimization procedure. Beginning with the kernel density estimate (KDE) from Eq. (81) and Eq. (82), the bandwidth σ of the Gaussian functions is needed. The bandwidth impacts the resolution of the KDE. While σ could be set by the user or estimated as part of the registration, we found that setting it a factor of the minimal pairwise distance

$$\sigma = \min(\frac{1}{L}\sum_{i=1}^{L}\min_{j}||d_i - d_j||\,, \frac{1}{K}\sum_{k=1}^{K}\min_{l}||m_k - m_l||), \tag{144}$$
$$\forall\, j \in [1:L]\,,\ i \neq j \ \text{ and } \ \forall\, l \in [1:K]\,,\ k \neq l$$

sufficed. Also, in the covariance matrix Q_i specifying the noise model u_k in Eq. (108) must be defined. We chose

$$Q_0 = \operatorname{argmin}\Big(\Big[.01, \frac{\pi}{90}, \frac{\pi}{90}, \frac{\pi}{90}, .01, .01, .01\Big]\Big) \tag{145}$$

as the covariance matrix for optimizing the similarity transformation parameters and

$$Q_i = \operatorname{argmin}([.01, .01, .01, .005, .005, .005, .001]) \tag{146}$$

as the covariance matrix used in the optimization of the GRBFs' variables, where $i \in [1, N]$. These dynamics make the assumption that the point sets being registered have been scaled to lie in the unit cube (for improved numerical conditioning) but can be re-scaled to fit data set with other diameters. All experiments are performed with the above values for σ, Q_0, and Q_i, $i \in [1, N]$.

4.4.2 Evaluating the Injectivity Constraint

Smoothness of a deformation can be enforced by checking that, in the domain S, $\forall \vec{x} \in S$ the condition

$$G_{lb} < d_J(\vec{x}) \tag{147}$$

holds. If the greatest lower bound of $d_J(\vec{x})$ is $G_{lb} = 0$, from Eq. (128), the deformation is injective everywhere. A smoother deformation can be found by setting G_{lb} to a value greater than zero and requiring that Eq. (147) holds. A related problem was considered in [16]: finding all of the maxima of a Gaussian mixture. In this work, the minimum of the derivative of a Gaussian mixture is desired for checking Eq. (147), which can be seen from Eq. (149)-(151). Furthermore, the global minimum only is required.

The scaled, rigid transformation in Eq. (85) can be represented as a matrix

$$A = \begin{bmatrix} s \cdot \cos(\theta) & -s \cdot \sin(\theta) \\ s \cdot \sin(\theta) & s \cdot \cos(\theta) \end{bmatrix}. \tag{148}$$

Then, for the two dimensional case,

$$d_J(\vec{x}) = \frac{\partial \mathcal{L}_x}{\partial x}\frac{\partial \mathcal{L}_y}{\partial y} - \frac{\partial \mathcal{L}_x}{\partial y}\frac{\partial \mathcal{L}_y}{\partial x} \tag{149}$$

$$\frac{\partial \mathcal{L}_x}{\partial x} = s \cdot \cos(\theta) - \sum_{i=1}^{N} \frac{w_{x_i}}{\sigma_i^2}(x - \mu_{x_i}) e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2} \tag{150}$$

$$\frac{\partial \mathcal{L}_x}{\partial y} = s \cdot \sin(\theta) - \sum_{i=1}^{N} \frac{w_{x_i}}{\sigma_i^2}(y - \mu_{y_i}) e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2} \tag{151}$$

and $\frac{\partial \mathcal{L}_y}{\partial x}, \frac{\partial \mathcal{L}_y}{\partial y}$ can be computed similarly to Eq. (150)-(151). An analytic solution for the minimum of $d_J(\vec{x})$ in Eq. (149) does not exist. Instead, a numerical approach for finding the minimum is presented. While the 2D case is described in detail, the algorithm similarly extends to higher dimensions.

Minimization of $d_J(\vec{x})$ is presented in Algorithm 3: it is performed iteratively by dividing an initial domain S and bounding the function over the subdomains. From the symmetry of Gaussian basis functions, it is known that the minimum of the mixture will lie in the convex hull of the maxima of the individual Gaussian partial derivatives that compose the mixtures in Eq. (150)-(151).

Algorithm 3 Algorithm to bound $det(J(\mathcal{L}(\vec{x};\Theta)))$

1: **function** TEST_GLB(Θ, ϵ, G_{lb}) ▷ find the greatest lower bound of $d_J(\vec{x})$
2: $S = \{\Omega_0\}$ ▷ Ω_0 = convex hull of all derivative of Gaussian peaks
3: bdRange $= 2\epsilon$
4: **while** bdRange $> \epsilon$ **do** ▷ Run until can approximate $\det()$ within ϵ everywhere
5: $N = numberOfElements(S)$
6: **for** i=1:N **do**
7: $\Omega_c = S(i)$
8: $S = S \backslash \{\Omega_c\}$ ▷ Remove Ω_c from the list
9: $[U_{\Omega_i}, L_{\Omega_i}]$=computeBounds($\Omega_c, \Theta$)
10: **if** $(U_{\Omega_i} <= G_{lb}) \mid$ ($(L_{\Omega_i} \leq G_{lb})$ & $(U_{\Omega_i} - L_{\Omega_i} \leq \epsilon)$) **then**
11: return 1 ▷ Zero crossing detected or within ϵ of crossing
12: **else**
13: $[\Omega_{N+1}, ..., \Omega_{N+4}] = splitDomainUniformly(\Omega_c)$ ▷ Split domain into 4 subdomains
14: $S = \{S, \Omega_{N+1}, ..., \Omega_{N+4}\}$ ▷ Add to the list
15: bdRange=$\max_{\Omega \in S} (U_\Omega - L_\Omega)$
16: **if** $S == \emptyset$ **then** ▷ No subdomain contains zero crossing, $\det() > 0$
17: return 0

The algorithm initializes the list S containing subdomains of *interest* to the convex hull Ω_0 of all maxima/minima of the components of mixtures in Eq. (149). At each iteration, in line 9, the upper bound U_{Ω_i} and lower bound L_{Ω_i} for d_J are computed for each subdomain in S. If the current sudomain Ω_c cannot be ruled out (*i.e.*, Eq. (147) may be violated in Ω_c), Ω_c is subdivided evenly into four subdomains, line 13, for further consideration. If a subdomain is found to violate Eq. (147) or be within ϵ of doing so, the function exits with a *true* value. Otherwise, the test terminates with a *false* value in two cases: S is empty, line 16, and no violations were found, or no violations are found and d_J has been bounded within ϵ, line 15.

Bounding d_J by U_{Ω_c} and L_{Ω_c} over $\Omega_c \subset \Omega_0$ involves making several conservative estimates. First, we bound

$$\begin{aligned} &\min_{\vec{x}\in\Omega_c} \left(\frac{\partial \mathcal{L}_x}{\partial x} \frac{\partial \mathcal{L}_y}{\partial y} \right) - \max_{\vec{x}\in\Omega_c} \left(\frac{\partial \mathcal{L}_x}{\partial y} \frac{\partial \mathcal{L}_y}{\partial x} \right) \\ &\leq d_J(\vec{x}) \leq \\ &\max_{\vec{x}\in\Omega_c} \left(\frac{\partial \mathcal{L}_x}{\partial x} \frac{\partial \mathcal{L}_y}{\partial y} \right) - \min_{\vec{x}\in\Omega_c} \left(\frac{\partial \mathcal{L}_x}{\partial y} \frac{\partial \mathcal{L}_y}{\partial x} \right) \quad \text{and} \end{aligned} \tag{152}$$

$$\min_{\vec{x}\in\Omega_c}\left(\frac{\partial\mathcal{L}_x}{\partial x}\frac{\partial\mathcal{L}_y}{\partial y}\right) = \min\{ac, ad, bc, bd\} \text{ where} \tag{153}$$

$$a = \min_{\vec{x}\in\Omega_c}\left(\frac{\partial\mathcal{L}_x}{\partial x}\right) \quad b = \max_{\vec{x}\in\Omega_c}\left(\frac{\partial\mathcal{L}_x}{\partial x}\right) \text{ and} \tag{154}$$

$$c = \min_{\vec{x}\in\Omega_c}\left(\frac{\partial\mathcal{L}_y}{\partial y}\right) \quad d = \max_{\vec{x}\in\Omega_c}\left(\frac{\partial\mathcal{L}_y}{\partial y}\right) . \tag{155}$$

The remaining three parts of Eq. (152) are computed similarly to Eq. (153). Then, the partial derivatives $\frac{\partial\mathcal{L}_x}{\partial x}, \frac{\partial\mathcal{L}_y}{\partial y}, \frac{\partial\mathcal{L}_x}{\partial y}, \frac{\partial\mathcal{L}_y}{\partial x}$ are bounded to find $\min_{\vec{x}\in\Omega_c}(\cdot)$ and $\min_{\vec{x}\in\Omega_c}(\cdot)$ in Eq. (153)-Eq. (155) :

$$\begin{aligned} & s\cdot\cos\theta - \sum_{i=1}^{N}\max_{\vec{x}\in\Omega_c}\left(\frac{w_{x_i}}{{\sigma_i}^2}(x-\mu_{x_i})\cdot e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right) \\ & \leq \frac{\partial\mathcal{L}_x}{\partial x} \leq \\ & s\cdot\cos\theta - \sum_{i=1}^{N}\min_{\vec{x}\in\Omega_c}\left(\frac{w_{x_i}}{{\sigma_i}^2}(x-\mu_{x_i})\cdot e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right) . \end{aligned} \tag{156}$$

Further, the value $\min_{\vec{x}\in\Omega_c}(\cdot)$ in Eq. (156) is computed as

$$\begin{aligned} & \min_{\vec{x}\in\Omega_c}\left(\frac{w_{x_i}}{{\sigma_i}^2}(x-\mu_{x_i})\cdot e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right) \\ & = \begin{cases} \frac{1}{{\sigma_i}^2}\min_{\vec{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i})\cdot\max_{\vec{x}\in\Omega_c} e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2} & \text{if } \min_{\vec{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i}) \leq 0 \\ \frac{1}{{\sigma_i}^2}\min_{\vec{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i})\cdot\min_{\vec{x}\in\Omega_c} e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2} & \text{if } \min_{\vec{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i}) > 0 \end{cases} \end{aligned} \tag{157}$$

and the computation is similar for $\max_{x\in\Omega_c}(\cdot)$ of Eq. (156). The bounds on $\min_{\vec{x}\in\Omega_c}(\cdot)$ of Eq. (157) (and similarly on $\max_{\vec{x}\in\Omega_c}(\cdot)$) are

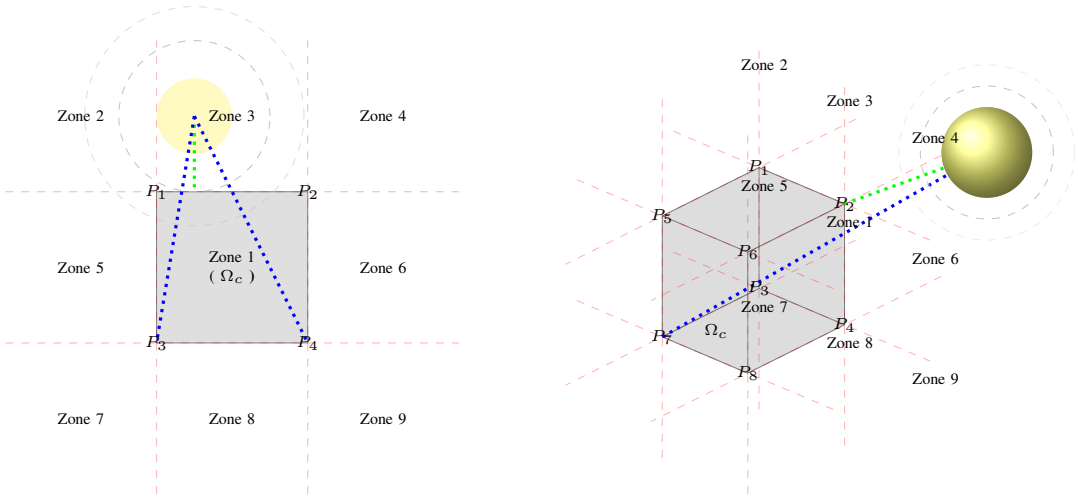
$$\min_{\vec{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i}) = \begin{cases} w_{x_i}\cdot\max_{\vec{x}\in\Omega_c}(x-\mu_{x_i}) & \text{if } w_{x_i} \leq 0 \\ w_{x_i}\cdot\min_{\vec{x}\in\Omega_c}(x-\mu_{x_i}) & \text{if } w_{x_i} > 0 . \end{cases} \tag{158}$$

To determine the bounds for $\min_{\vec{x}\in\Omega_c} e^{(\cdot)}$ and $\max_{\vec{x}\in\Omega_c} e^{(\cdot)}$ from Eq. (157) and compute Eq. (158), the domain is broken into zones as shown in Fig. 27; the corner points $P_j = \left[P_{j_x}, P_{j_y}\right]$, $j \in \{1, ..., 4\}$ determine the boundaries. Now, the quantities in Eq. (158) can be bounded as

$$\begin{aligned} & \min_{\vec{x}\in\Omega_c}(x-\mu_{x_i}) \geq P_{1_x}-\mu_{x_i} \qquad \max_{\vec{x}\in\Omega_c}(x-\mu_{x_i}) \leq P_{2_x}-\mu_{x_i} \\ & \min_{\vec{x}\in\Omega_c}(y-\mu_{y_i}) \geq P_{3_y}-\mu_{y_i} \qquad \max_{\vec{x}\in\Omega_c}(y-\mu_{y_i}) \leq P_{1_y}-\mu_{y_i} . \end{aligned} \tag{159}$$

Zone #	$\min_{x\in\Omega}\left(e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right)$	$\max_{x\in\Omega}\left(e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right)$
1	$\min_j e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2},\ j\in\{1,...,4\}$	1
2	$e^{-\frac{1}{2\sigma_i^2}\|P_4-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_1-\mu_i\|^2}$
3	$\min_j e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2},\ j\in\{3,4\}$	$e^{-\frac{1}{2\sigma_i^2}\|P_{0y}-\mu_i\|^2},\ P_{0y}=[0,P_{1_y}]$
4	$e^{-\frac{1}{2\sigma_i^2}\|P_3-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_2-\mu_i\|^2}$
5	$\min_j e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2},\ j\in\{2,4\}$	$e^{-\frac{1}{2\sigma_i^2}\|P_{0x}-\mu_i\|^2},\ P_{0x}=[P_{1_x},0]$
6	$\min_j e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2},\ j\in\{1,3\}$	$e^{-\frac{1}{2\sigma_i^2}\|P_{0x}-\mu_i\|^2},\ P_{0x}=[P_{2_x},0]$
7	$e^{-\frac{1}{2\sigma_i^2}\|P_2-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_3-\mu_i\|^2}$
8	$\min_j e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2},\ j\in\{1,2\}$	$e^{-\frac{1}{2\sigma_i^2}\|P_{0y}-\mu_i\|^2},\ P_{0y}=[0,P_{3_y}]$
9	$e^{-\frac{1}{2\sigma_i^2}\|P_1-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_4-\mu_i\|^2}$

Table 1: Bounds on a Gaussian function over the domain Ω_c in 2D. The bounds depend on which zone relative to Ω_c , from Fig. 27(a), the function is located in.



(a) Zones in 2D with respect to Ω_c (Zone 1).

(b) Zones in 3D; there are 27 total but only the first 9 are labeled.

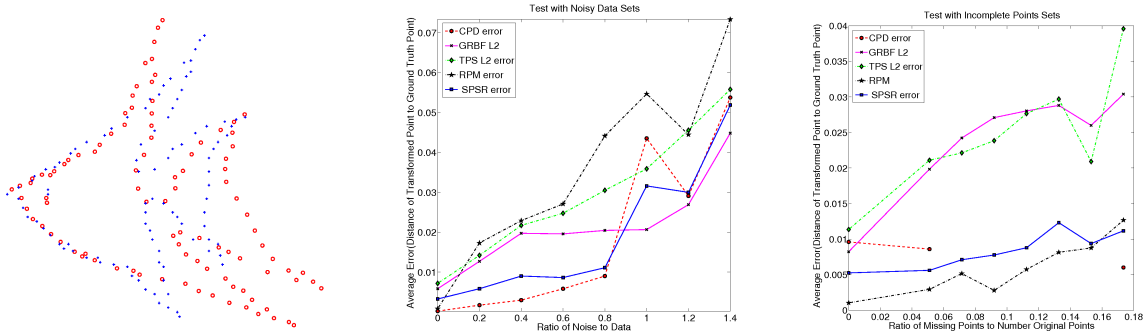
Figure 27: To find bounds for derivative of Gaussian functions over Ω_c (shown in gray) the 2D/3D space is divided into zones. The circle/sphere show the center of a sample Gaussian function. Clearly, the minimum for this function is located at a point at the end of a blue line and the maximum at the end of the green line.

Finally, bounds on the Gaussian functions within Ω_c are computed using Table 1. This table is made by examining Fig. 27(a). For example, if a Gaussian function is located in Zone 3, its maximum value is located at the boundary of the green line and Ω_c and the minimum is either at P_3 or P_4. All possible scenarios for the location of a Gaussian function and the corresponding bounds are listed in Table 1.

4.5 Results

4.5.1 Examples in 2D

Noisy and Incomplete Data 2D The first set of tests performed are under controlled conditions. They demonstrate the ability of the proposed algorithm to handle missing points where no correspondences exist and noisy data sets with additional points not belonging to the structures of interest. The ground truth data was generated by taking a model point set (from `http://www.cise.ufl.edu/~anand/students/chui/research.html`) and deforming it to create an ideal data set, both depicted in Fig. 28(a). The test data was created by adding varying levels of noise as presented in Fig. 29(a) or by

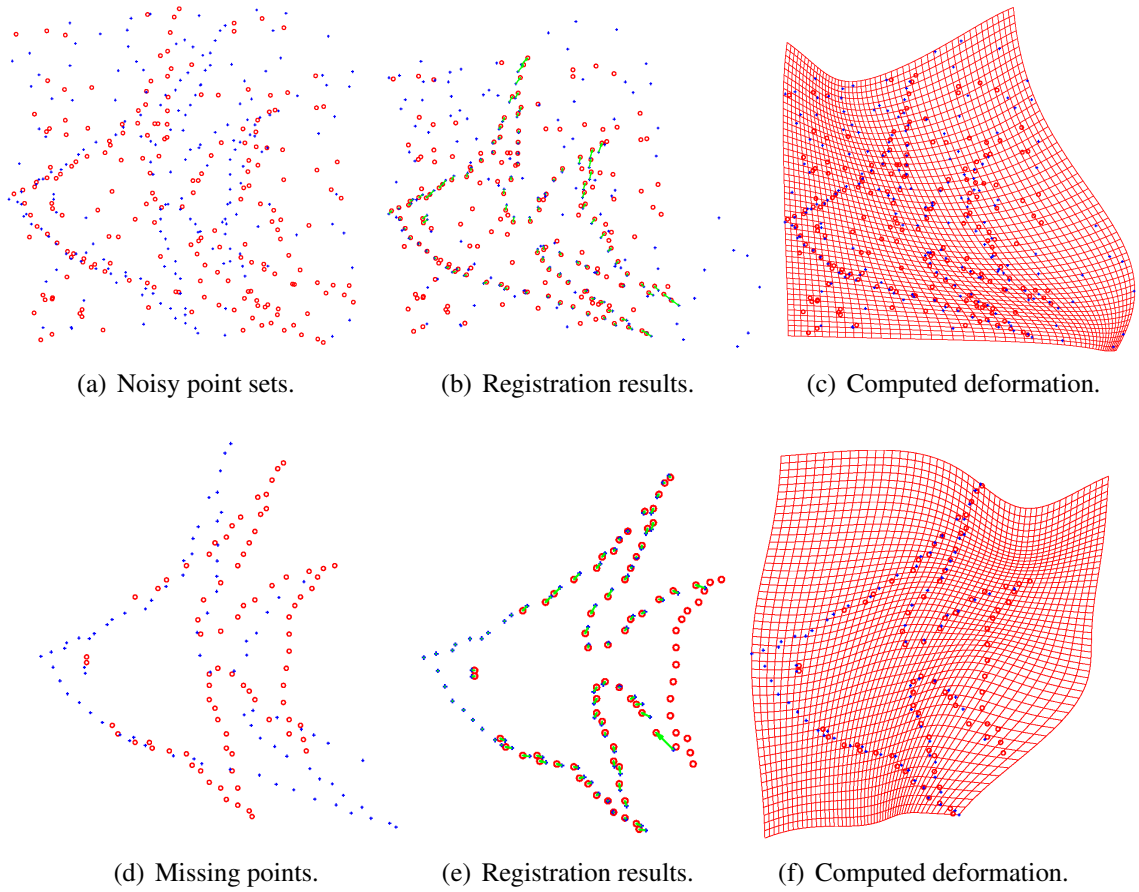


(a) Original point sets.

(b) Average error per point for noisy data sets.

(c) Average error per point for incomplete data sets.

Figure 28: Fish Data: Plot of the original data points and error plots for data corrupted by noise and data with missing correspondences. SPSR is the approach proposed in this work; its is compared to CPD [66] , RPS [10] (GRBF L2 and TPS L2), and RPM [23]. A parameter sweep is performed for CPD, RPS, and RPM and settings corresponding to the lowest error while producing a one-one mapping are used to generate the graphs. In Fig. 28(c), the graph for CPD has missing data points because it failed to find a non-overlapping mapping for all 56 parameter settings.



(a) Noisy point sets. (b) Registration results. (c) Computed deformation.

(d) Missing points. (e) Registration results. (f) Computed deformation.

Figure 29: The synthetic examples that were used to produce Fig. 28 are shown. The starting point sets are in Fig. 28(a). These sets were corrupted by random noise; Fig. 29(a) shows an example with noise to data ratio of 1.2 . In the other test, one of the sets had points removed from the front and the other from the back; Fig. 29(d) shows an example where 18% of points were removed from each set. Registration results are shown and the errors we measure represented by green arrows.

removing points from each of the two sets as in Fig. 29(d). The baseline algorithms we used for comparison are: CPD [66] , RPS [10], and RPM [23] (discussed in Section 4.1). For RPS, the authors proposed two parametrizations for the deformation: TPS and GRBFs, labeled 'TPS L2' and 'GRBF L2', respectively, in Figs. 28(b) and 28(c).

For CPD, the user selects values for two parameters: the regularization weight λ and the Gaussian basis width β. For RPS using TPS, the user sets λ and σ, the width of the kernel in Eq. (81)-(82) . For RPS using GRBFs, λ, β, σ must be chosen. Finally, for RPM the initial temperature T_0 and the annealing rate r must be set. In our experience, these

parameters had to be adjusted for each noise level and missing point level to obtain the best registration. Since it is not obvious what the settings should be, a parameter sweep was performed for each of the competing approaches using 56 parameter pairs. 'GRBF L2' has three parameters, so 392 parameter triplets were tried per registration. Thus, the best possible results are reported in Fig. 28 for the competing approaches. Outside of the constants set in Section 4.4.1 and the number of particle and iterations for the particle filter (defined once and used for all experiments), the stochastic point set registration (SPSR) algorithm has just one parameter: the number of basis functions. The user only has access to this parameter and changing it has the intuitive effect of improving registration accuracy, explained in Section 4.3.4, at the cost of increased computation time. For all experiments, $N = 10$, which was empirically determined as reasonable compromise between running time and accuracy.

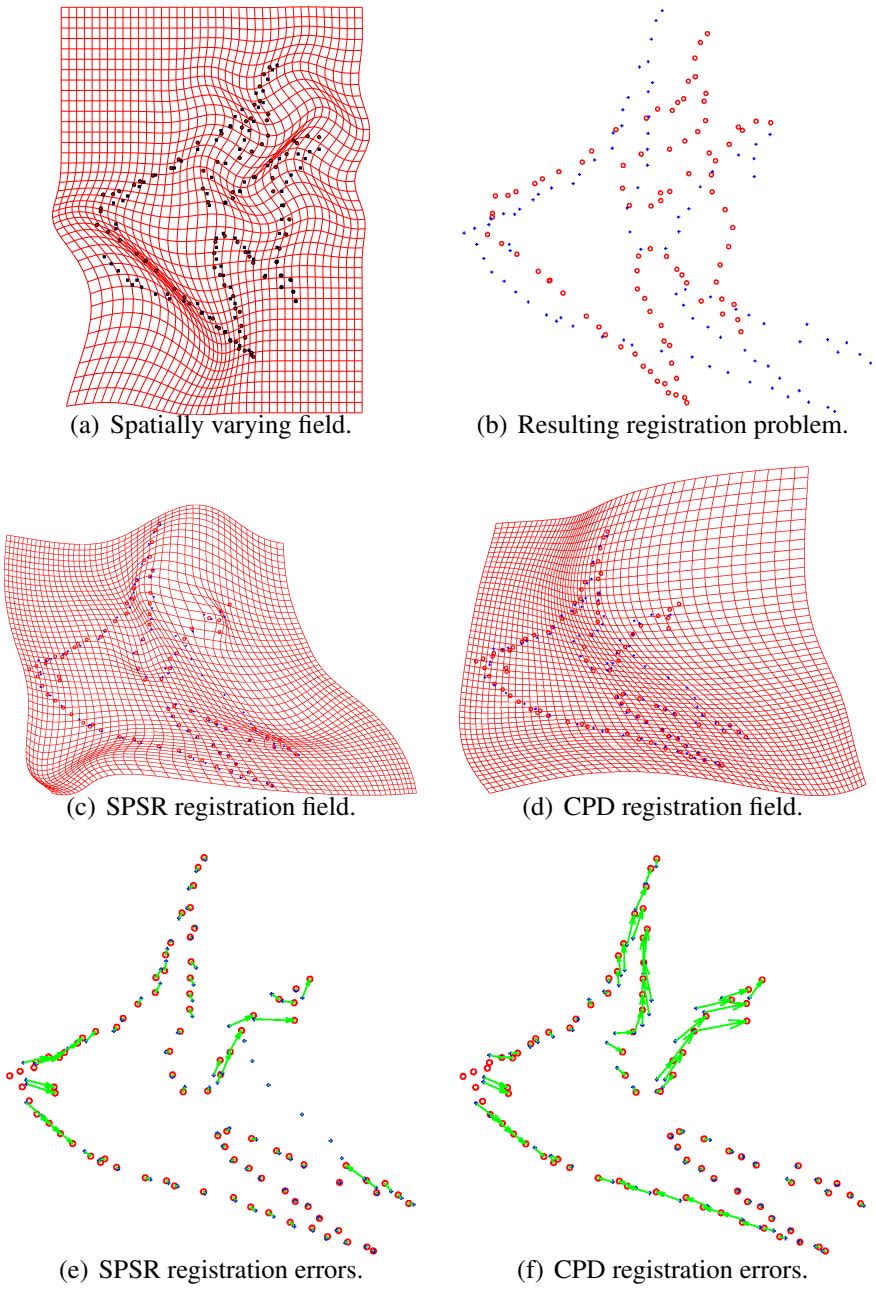
The plot of error as a function of noise level, depicted in Fig. 28(b), was generated by creating ten synthetic examples at each noise level plotted, performing the registration, then plotting the average error rate of the ten trials. Similarly, the error as a function of the missing points ratio, in Fig. 28(c), was generated by removing a certain percentage of points from each data set and computing the registration error. Given the ground truth, the error is computable by taking the average of the Euclidean distance between all points and their correspondences in the stationary set (error vectors are shown as green arrows in Fig. 29(b) and Fig. 29(e)).

The SPSR performs favorably compared to the competing algorithms in both tests while at the same time maintaining a non-overlapping deformation field Eq. (128) (not guaranteed by any of the competing approaches). In a real application, the competing approaches will likely not achieve the theoretical performance shown in Fig. 28 because ground truth is not available, and requiring a human to perform a parameter sweep and select the best of 56 registrations is time consuming and is not trivial.

Deformation Varying in Spatial Frequency This experiment demonstrates the difficulty with regularizing based on global smoothness of the field. A template from the partial data experiment, as in Fig. 29(d), is the starting point; then, a deformation with spatially varying frequencies is applied to one of the templates. This deformation is shown in Fig. 30(a). Thus, regularization is needed to prevent overlaps in the field, but penalizing smoothness globally does not allow an algorithm to capture high frequency deformations. Since, as previously noted, the best setting for RPS based approaches turned out to be $\lambda = 0$ for these experiments and the regularization has no effect, this test is performed for the CPD approach only. The registration results for CPD and SPSR are shown in Fig. 30. The effect of global regularization described in this paragraph is particularly evident in the regions around the top fin of the fish; here, CPD commits its largest errors while SPSR successfully captures the high frequency components of the deformation while maintaining an injective field.

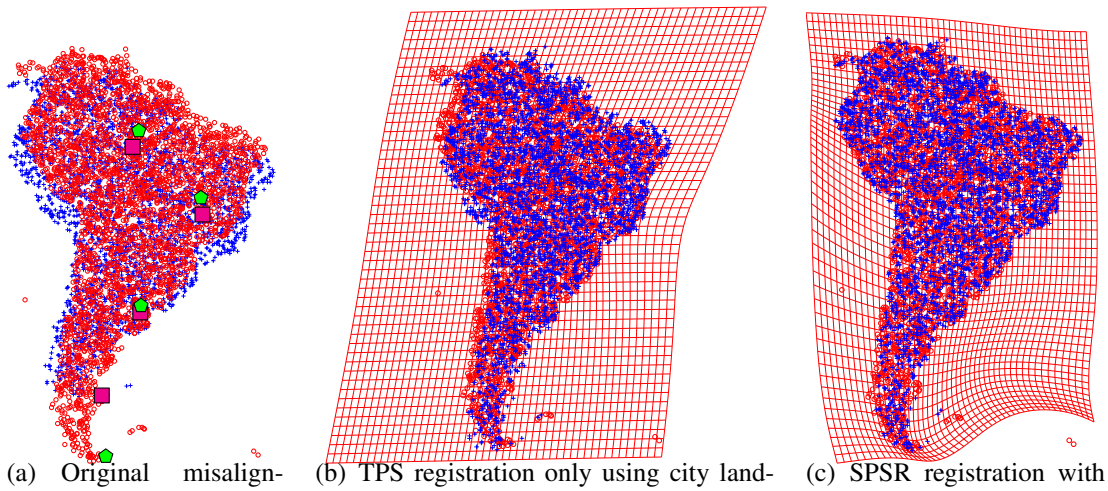
Landmark Constrained 2D Example This section demonstrates an example in 2D when a few user constraints greatly simplify the registration problem. The original, misaligned point sets are depicted in Fig. 31(a). Two maps were created by projecting the earth's surface onto a plane according to the Lambert and Mercator projections. The resulting shape of the South American continent differs widely; to create the point sets, images of South America, from the two projections, were sampled uniformly. Additionally, the coordinates of the following four cities, denoted by pentagons and squares, were marked in each image: Brasilia, Buenos Aires, Manaus, and Ushuaia. Thus, the registration problem consists of aligning the point sets subject to the condition that location of the cities match exactly.

The first step of registration establishes a rough alignment using just the four pairs of corresponding cities. Using the known correspondences, an optimal interpolating thin plate spline (TPS) was computed. Then, the point sets were aligned using the TPS deformation field. The results in Fig. 31(b) clearly show that four correspondences are insufficient



(a) Spatially varying field.
(b) Resulting registration problem.
(c) SPSR registration field.
(d) CPD registration field.
(e) SPSR registration errors.
(f) CPD registration errors.

Figure 30: A template with points missing, similar to the one in Fig. 29(d) was deformed according to the field shown in Fig. 30(a). This example shows a deformation field exhibiting spatially varying smoothness. A low frequency perturbation is present in the bottom, left and a high frequency component deform the upper, right portion of the sub-domain. While performing the registration, a Tikhonov regularization, used in CPD and RPS, based on the assumption of a globally smooth field, prevents the high frequency component from being recovered. This figure compares the fields produced by SPSR and CPD. The global smoothness penalty imposed by CPD prevents the algorithm from capturing high frequency components of the deformation in Fig. 30(d), in the upper right of the domain. Consequently, largest errors are committed here. The implicit regularization approach of SPSR allows it to capture a majority of these deformations.



(a) Original misalignment.

(b) TPS registration only using city landmarks.

(c) SPSR registration with landmark constraints.

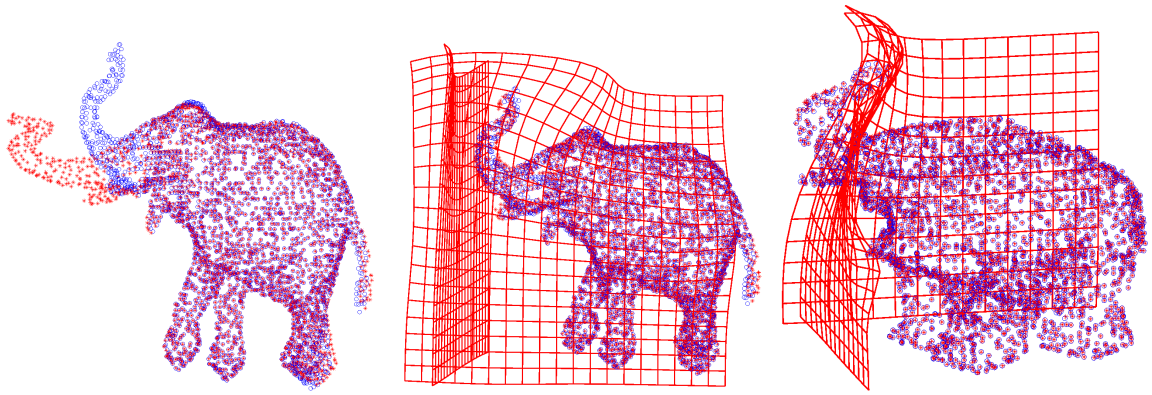
Figure 31: An example of constrained registration in 2D is shown here. Two misaligned point sets, along with four corresponding pairs of points are seen in Fig. 31(a). A rough alignment is performed using TPS and the four pairs of corresponding points, shown in Fig. 31(b). The landmarks are aligned and constrained not to move during the final step. Results of doing point set registration subject to known points remaining stationary are in Fig. 31(c).

to align the point set exactly; however, the gross alignment errors have been corrected. At this point, it is known that the location of the cities match; consequently, using the approach in Section 4.3.1 these points are fixed for the remainder of the registration process. Subject to these constraints, the SPSR algorithm is applied to achieve the final registration in Fig. 31(b).

4.5.2 Examples in 3D

Synthetic Example The extension of SPSR to 3D is straightforward; it involves a mild increase in the optimization space of three degrees of freedom for the rigid parameters and two degrees of freedom per Gaussian basis. Since the increase is moderate, we expect similar performance in 3D as for 2D.

In the first experiment, a point cloud of an elephant was generated with 3,093 points and was deformed to create the two point clouds in Fig. 32(a). The deformation is large in magnitude and twists the trunk of the elephant in 3D. The local nature of the transformation



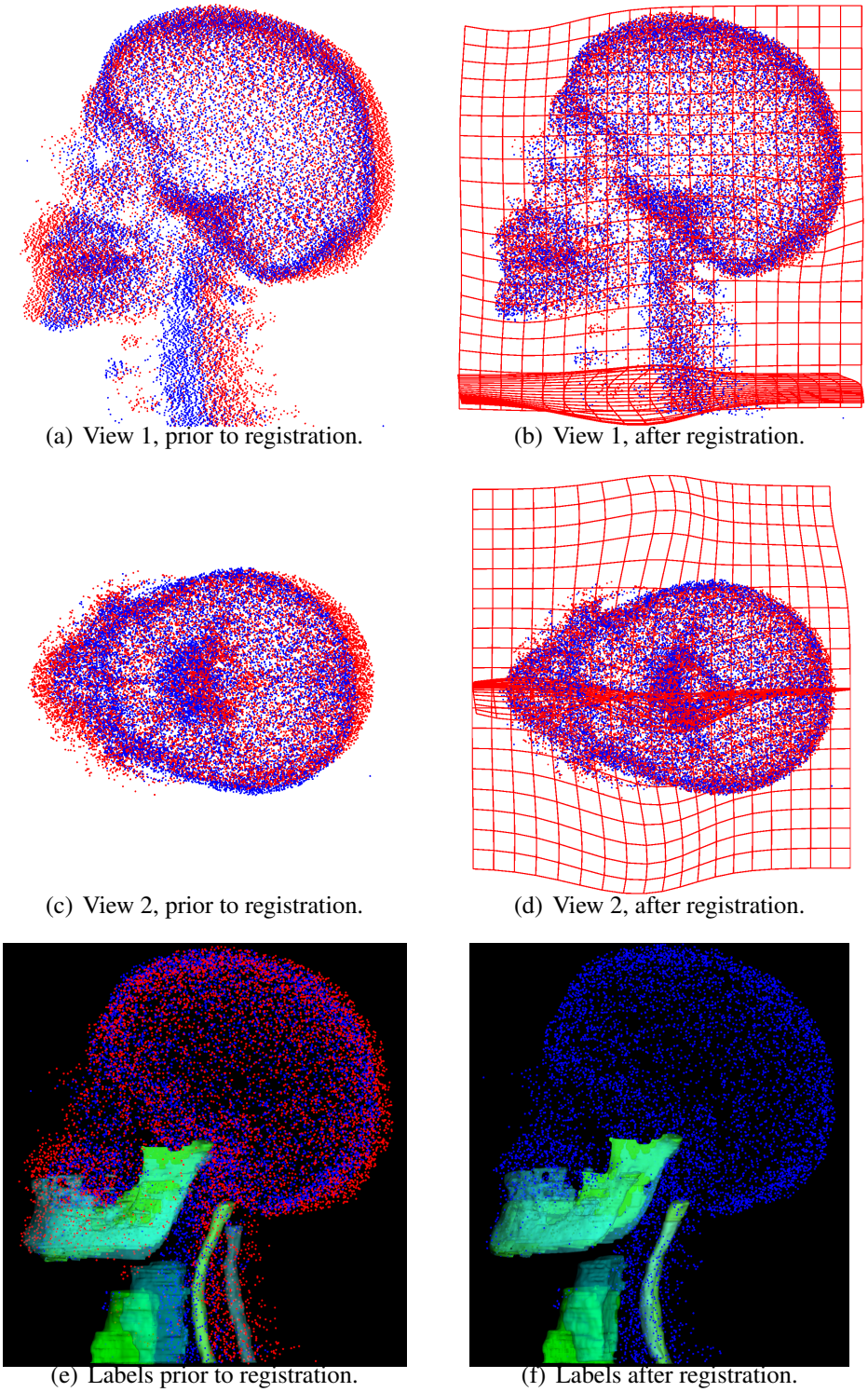
(a) Elephant point sets. (b) Registration view 1. (c) Registration view 2.

Figure 32: A 3D point cloud of an elephant was deformed by applying a known deformation to create a synthetic data set. The sets are composed of 3,093 points. In Fig. 32(b) and 32(c), the computed deformations are not self-overlapping and are local around the misaligned points, as desired.

was intentional. We see that the SPSR algorithm correctly focuses on the region around the trunk and only deforms space in this region without altering far away areas.

Real 3D Example For the second experiment, computed tomography (CT) image volumes of two different patients were obtained. With a simple threshold, bone label maps were extracted; these labels were then uniformly sampled to generate point clouds consisting of 14,832 and 14,723 points. The initial point sets are shown in Fig. 33(a), and despite a rigid alignment, significant differences exist. Here, the target application is atlas based segmentation (i.e., align the CT volumes using just the sampled point clouds and transfer label maps from the reference patient to the incoming patient). The results of registration are shown in Fig. 33. In this example, ground truth label maps for the mandible, larynx, and spinal cord are available for both patients, in Fig. 33(e). Using the computed deformation fields from the point set registration, the label maps of the reference image were deformed. Clearly, in Fig. 33(f), the organ labels are well aligned.

Landmark Constrained 3D Example Despite accurate registration of the bony structures and organs around them, in areas where no information was available (i.e., no points



(a) View 1, prior to registration.

(b) View 1, after registration.

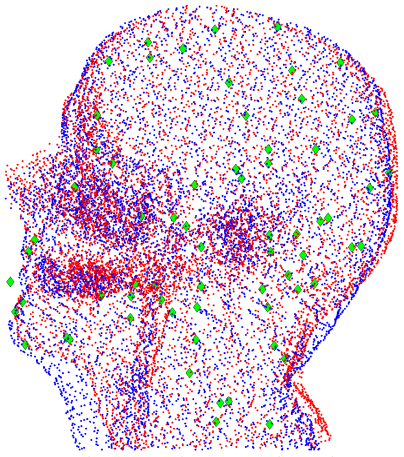
(c) View 2, prior to registration.

(d) View 2, after registration.

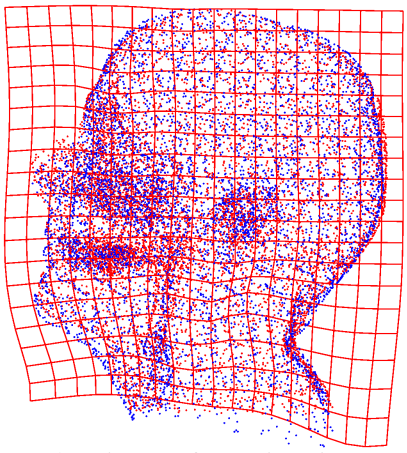
(e) Labels prior to registration.

(f) Labels after registration.

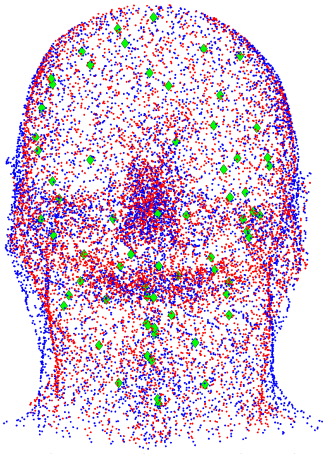
Figure 33: Real point clouds (14,832 and 14,723 points) generated from CT volumes of different patients. Clearly, a rigid registration is insufficient as there is large variation in the tilt of the heads (vertebral columns don't align), the shape of the skull, and the shape of the jaws. Medical image registration is the targeted application, and a non-overlapping field is particularly important.



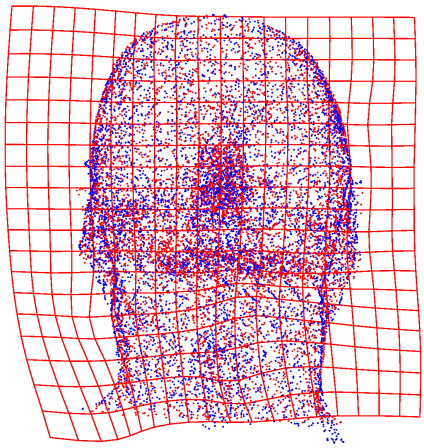
(a) View 1, prior to registration.



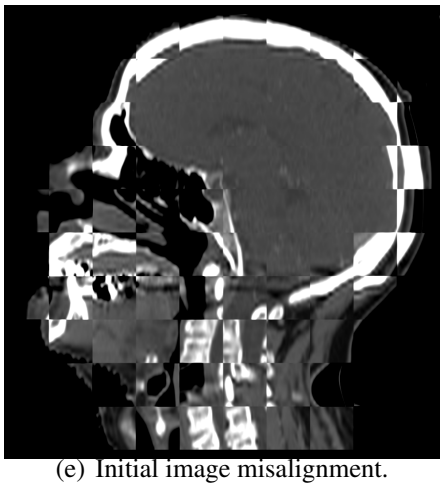
(b) View 1, after registration.



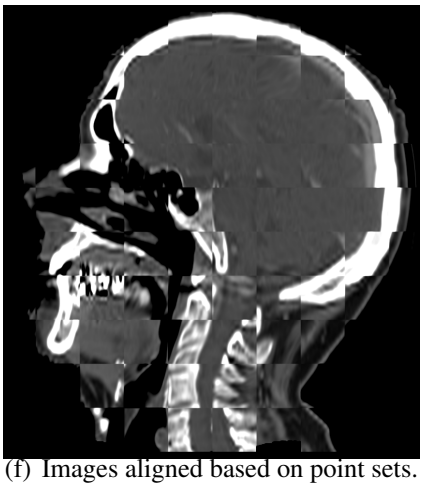
(c) View 2, prior to registration.



(d) View 2, after registration.



(e) Initial image misalignment.



(f) Images aligned based on point sets.

Figure 34: Point clouds of 6,842 and 6,877 points were generated from CT volumes by sampling the label maps of patients' flesh. Constrained registration was performed to align the point clouds representing flesh while keeping points on the registered skeletons from Fig. 33 stationary. The original misalignment and the 67 constraint points marked with green diamonds are displayed in Fig. 34(a) and Fig. 34(c). Finally, the deformations from Fig. 33 and Fig. 34 were sequentially applied to the original CT volume. The starting misalignment is seen in Fig. 34(e) and the result of the two step registration is in Fig. 34(f).

present to guide the registration), there is no guarantee that the original CT image will be aligned. A second stage of registration is performed to align these regions by first warping the initial CT volume according to the deformation computed in Fig. 33. At this stage, a label map of the patients' flesh was made, again using a simple threshold operation. The point clouds that are a result of sampling the masks are shown in Fig. 34(a) and Fig. 34(c).

This problem is complicated by the fact that registering the flesh point clouds would likely undo the alignment of the bone achieved in Fig. 33. We can mitigate this complication by introducing a set of constraints, as in Section 4.3.1. The constraint points are chosen as a subset of the aligned skeleton point clouds and do not move at all during the subsequent registration. Thus, regions belonging to the bone tissue are restricted in their motion. To generated the deformation field in Fig. 34, 67 points serve to constrain the problem. Subject to the constraints, registration is performed and the deformation field in Fig. 34(b) and Fig. 34(d) are computed. Finally, the originally misaligned CT volumes from Fig. 34(e) are registered using a composititon of the two deformation fields from Fig. 33-34 to obtain the registration visualized in Fig. 34(f).

4.6 Chapter Conclusion

This chapter presented a stochastic registration methodology and demonstrated it on registration of point sets. The resulting registration framework was robust for scenes with noise and missing points; it can be applied to points clouds of arbitrary dimension. The deformation field $\mathcal{L}$ consisted of the additive composition of a rigid transformation and a non-rigid transformation through a GRBF network. The key parameter, N (the number of basis functions), can be increased to improve the registration with increased run time being the only detrimental effect. The effect of increasing N is intuitive since it simply allows the algorithm to treat increasingly finer deformations as opposed to a smoothness penalty employed by state of the art point set registration algorithms (e.g., RPS and CPD).

Additionally, important constraints on the transformation $\mathcal{L}$ are imposed to ensure physically realistic deformations. This last point is essential for medical imaging and other applications that obtain point sets from physical objects. As part of the implementation for the registration framework, a numerical method for evaluating the overlap constraint was presented.

CHAPTER V

STOCHASTIC APPROACH FOR LARGE-DEFORMATION IMAGE REGISTRATION

In a number of applications, image registration is essential because all information is retained, in contrast to aligning representative point sets, for instance, which attempt to form a sparse representation for objects. If the chosen metric is sufficient to align the images, successful registration can be expected. Despite having a correct global optimum, however, the cost function is likely to suffer from the presence of a large number of local optima, which lead to erroneous registration. Large initial misalignment of images (i.e., poor initialization for the algorithm) is the culprit in these failure scenarios. This problem, in particular compensating for large non-rigid deformations by searching for a global optimum stochastically, will be explored in this chapter. In Section 5.1 a literature review is provided. Section 5.2 introduces preliminary concepts that are useful for Section 5.3. Section 5.3 presents the stochastic image registration (SIR) algorithm. Performance of the algorithm on 2D and 3D data sets is presented in Section 5.4.

5.1 Existing Work

Non-rigid registration algorithms naturally split into two classes: parametric models and physical models. Approaches falling into the first category use basis functions to approximate the true deformation fields and optimize over the parameters in this representation. Physical models impose properties on the deformation field according to the type of deformation that is expected to be seen.

In the category of parametric models, Rueckert *et.al* [81] proposed placing anchor points for B-splines uniformly within the image domain. Optimization on the B-spline

parameters is done via gradient-descent from coarse to fine resolution. Regularity on the deformation field is imposed by penalizing the smoothness of the deformation field globally. An extension of this approach to ensure injectivity of the deformation is proposed in [80]. The new implementation of this method is called *NiftyReg* and will be used for comparison in Section 5.4.

A number of physical models have been proposed for image registration. *Demons* [90] is a popular algorithm for image registration inspired by Maxwell's demons thought experiment in thermodynamics. The registration is seen as a diffusion process where the boundaries in the target image are considered to be semi-permeable membranes and the moving image diffuses through these boundaries to achieve alignment. A common variant of the demons approach places effectors at each pixel, uses optical flow to compute the diffusion force at each effector, deforms the image accordingly and iterates until convergence. This approach works well when deformations are small since computation of the optical flow vector relies on a linear model for intensity change. A diffeomorphic demons algorithm [97] exists while the original formulation did not enforce this constrains and regularization was achieved by Gaussian smoothing the field to preserve the topology.

Elastic registration can be generally written as the following optimization problem: $\underset{u}{\operatorname{argmin}}\ \Psi_s(u, I_0, I_1) + \lambda\Psi_r(u)$. A trade-off between maximizing Ψ_s, the similarity between images I_0 and I_1, is weighted by λ against the penalty for large deformations Ψ_r. Thus, large displacements are discouraged in this formulation. To remedy this effect, Christensen *et. al* proposed a viscous fluid flow model for large deformations in [21] and the general, corresponding registration problem is $\underset{v}{\operatorname{argmin}}\ \Psi_s(u, I_0, I_1) + \int \Psi_r^v(v)dt$. Here, the displacement field is the time integral of the time-varying velocities: $u = \int v\ dt$, which are regularized through Ψ_r to ensure u remains a homeomorphism but do not penalize large displacements. Avants *et. al* [6] extended this theory and proposed the symmetric image normalization *SyN* method. The authors formulate the registration problem symmetrically, with respect to the moving and stationary image, and compute the geodesic between the

images within the manifold of diffeomorphic mappings. We compare our approach to the *SyN* registration results in Section 5.4.

Since the focus of this work is on stochastic optimization, we discuss several algorithms of this nature. Two works have explored using a particle filter (PF) for image registration: Kasetkasem *et.al* [47] and Santanta *et.al* [5]. In [47], the authors perform rigid registration and experiment on two dimensional images. The authors present a non-rigid registration method in [5] that is composed of a PF for affine registration followed by an optical flow deformable alignment and alternate between the two steps until convergence. These approaches applied a stochastic search methods to capture significant rigid or affine deformations. However, even after an affine alignment, large non-linear deformations between images can exist that a local registration method such as optical flow fails to capture. A small sample of recent works in image registration was provided here, and the reader is referred to [108, 71, 39] for a more detailed literature review.

5.2 Preliminaries

5.2.1 Problem Statement

Given a moving image $I_D(\vec{x})$ and a target image $I_M(\vec{x})$, registration is the task of finding a mapping between the two images with the appropriate properties. We consider $2D$ images or $3D$ volumes, and the objective is to find a mapping $\mathcal{L} : \mathbb{R}^l \rightarrow \mathbb{R}^l$ that minimizes the distance, d, between the images I_M and I_D :

$$\min_{\mathcal{L}} d(I_M, I_D(\mathcal{L})) \; . \tag{160}$$

The measure of similarity $d(\cdot, \cdot)$ can be selected to fit the application but experiments are performed with the L_2 error

$$d(I_M, I_D(\mathcal{L})) = \int_{\Omega} \Big(I_M(\vec{x}) - I_D(\mathcal{L}(\vec{x})) \Big)^2 d\Omega \tag{161}$$

to generate the results presented in Section 5.4. This similarity metric is appropriate when I_M, I_D were taken by the same image sensor and the same objects have similar intensities

in both images.

5.2.2 Parametrizing the Displacement Function

The space of non-rigid transformations is infinite-dimensional, which complicates finding a solution to Eq. (160). Parametric registration algorithms using gradient descent for optimization (e.g., Rueckert *et.al* [81]) typically employ a multi-resolution framework that involves refining the basis spacing and increasing the working image resolution at each level of the registration. A gradient descent is performed to compute the optimal basis weights at each stage; however, for large deformations, even a hierarchical approach like this one is not sufficient, as demonstrated in Section 5.4, through experimental comparison.

To make the optimization tractable, we limit the degrees of freedom of the deformation field $\hat{\mathcal{L}}$ by defining it to be an additive composition of three deformations. The proposed parametrization for $\hat{\mathcal{L}}(\cdot)$ is

$$\begin{aligned}\hat{\mathcal{L}}(\vec{x};\Theta) &= \mathcal{L}_s(\vec{x};\Theta_s) + \mathcal{L}_g(\vec{x};\Theta_g) + \mathcal{L}_l(\vec{x};\Theta_l) \\ &= \underbrace{A\vec{x} + \vec{b}}_{similarity} + \underbrace{\sum_{i=1}^{N} \vec{w}_i \, \mathcal{N}\big(\vec{x} \,|\, \vec{\mu}_i, \sigma_i^2 I\big)}_{global} + \underbrace{\sum_{j=1}^{M} \vec{w}_j \, \mathcal{N}\big(\vec{x} \,|\, \vec{\mu}_j, \beta^2 I\big)}_{local} \; . \end{aligned} \tag{162}$$

In addition, $\hat{\mathcal{L}}$ must respect physical constraints: the mapping must be injective and orientation preserving. Enforcing the constraint in Eq. (128) will ensure an orientation preserving and one-to-one transformation. Thus, the constrained optimization problem that will be solved to register two images is

$$\begin{aligned} &\min_{\Theta} \; E(\Theta) \\ &\text{s.t. } d_J(\vec{x}) > 0 \qquad \text{where} \\ &E(\Theta) = d(I_M(\vec{x}), I_D(\hat{\mathcal{L}}(\vec{x};\Theta))) \; . \end{aligned} \tag{163}$$

One way to enforce the property that $\mathcal{L}_s, \mathcal{L}_g, \mathcal{L}_l$ correct for increasingly finer misalignments is to perform a sequential registration starting from global, moving to local and composing

the computed deformations to obtain the overall deformation:

$$\mathcal{L}(\vec{x}; \Theta_s, \Theta_g, \Theta_l) = (\mathcal{L}_l \circ \mathcal{L}_g \circ \mathcal{L}_s)(\vec{x}) \ . \tag{164}$$

Further, the optimization is performed sequentially as follows:

$$\hat{\Theta}_s = \min_{\Theta_s} R(\Theta_s | \Theta_g, \Theta_l) \tag{165}$$

$$\hat{\Theta}_g = \min_{\Theta_g} R(\Theta_g | \hat{\Theta}_s, \Theta_l) \tag{166}$$

$$\hat{\Theta}_l = \min_{\Theta_l} E(\Theta_l | \hat{\Theta}_s, \hat{\Theta}_g) + \lambda E_r(\mathcal{L}(\vec{x}; \Theta_l, \hat{\Theta}_s, \hat{\Theta}_g)) \tag{167}$$

$$\text{s.t. } d_J(\vec{x}) > 0$$

$R(\cdot)$ is defined in Section 5.3. In Eq. (167), $E_r(\cdot)$ is a regularizing term weighted by λ. For instance, in 2D, this term is

$$E_r(\mathcal{L}) = \int_x \int_y \left[\left(\frac{\partial^2 \mathcal{L}}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathcal{L}}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 \mathcal{L}}{\partial y^2} \right)^2 \right] dxdy \ . \tag{168}$$

Here, too, a clear, hierarchical alignment is seen: starting from a trivial initialization of Θ_g, Θ_l, an optimal Θ_s is computed in Eq. (165), then holding $\hat{\Theta}_s, \Theta_l$ constant, the best $\hat{\Theta}_g$ is found in Eq. (166), and finally optimization over Θ_l is performed in Eq. (167). The difference between the proposed method and existing multi-resolution approaches is that optimization in addition to the parametrization changes at each registration level: Eq. (165) and Eq. (166) will be solved stochastically to capture large misalignments, and for Eq. (167), a gradient descent approach is used because more degrees of freedom are in the representation of $\mathcal{L}_l$, which is necessary to capture finer registration misalignments. The deterministic registration allows more bases to be used, which would make a stochastic method too computationally intensive. Solutions to Eq. (165)-(167) are discussed in Sections 5.3.1-5.3.3, respectively.

5.3 Optimization Algorithm

5.3.1 Stochastic Optimization of $\mathcal{L}_s$

A stochastic method for finding the optimal similarity transformation parameters is discussed in this section. Optimization of a cost function $R(\Theta)$ is performed and two choices for $R(\Theta)$ are presented in Eq. (169)-(170) :

$$R_1(\Theta) = E(\Theta) \tag{169}$$

$$R_2(\Theta) = \begin{cases} E(\Theta) & \text{if } r(\cdot) > 0 \\ B \cdot (\|I_M(\vec{x})^2\|_{L^2}^2 + \|I_D(\vec{x}; \mathcal{L}(\vec{x}; \Theta))\|_{L^2}^2) & \text{if } r(\cdot) \leq 0 \end{cases} \tag{170}$$

where $r(\Theta) = \det(J(\mathcal{L}(\vec{x}; \Theta)))$.

First, for unconstrained optimization of Eq. (163), $R(\Theta) = R_1(\Theta)$ is chosen. Second, to preserve the orientation and enforce injectivity, $R(\Theta) = R_2(\Theta)$ is selected for optimization. In SIR, optimization of the function $R(\Theta)$ is performed using a particle filter, as described in Section 4.2.6. The cost function $R_2(\Theta)$, in Eq. (170), imposes a soft constraint on the injectivity and orientation of $\mathcal{L}$, which is discussed in Section 4.3.1.

The parameter vector for a similarity transformation is composed of quaternion coordinates $[w, x, y, z]$ for rotation and scale and a translation vector $[a, b, c]$:

$$\Theta_s = [w, x, y, z, a, b, c] \, . \tag{171}$$

In this note, the discussion assumes a 3D image volume (the 2D case is simpler) and $R(\Theta) = R_2(\Theta)$ is selected. The low dimensionality of Θ_s makes it possible to solve Eq. (165) using the PF method, directly. This approach allows large translations, scale changes, and rotations to be captured because the algorithm can escape local minima, which present problems for a gradient descent technique.

5.3.2 Stochastic Optimization of $\mathcal{L}_g$

It is likely that gross misalignments still exist after two images have been aligned using the procedure in Section 5.3.1. Hence, the next step is to solve Eq. (166) by computing a

global, non-linear, deformation, represented by Θ_g:

$$\begin{aligned} \Theta_g &= [\vec{\mu}_1, \vec{w}_1, \sigma_1, \ \cdots \ , \vec{\mu}_N, \vec{w}_N, \sigma_N] \\ \vec{\mu}_i &= [\mu_{x_i}, \mu_{y_i}, \mu_{z_i}] \quad \text{and} \\ \vec{w}_i &= [w_{x_i}, w_{y_i}, w_{z_i}] \ . \end{aligned} \tag{172}$$

A stochastic approach is preferred to robustly capture large deformations; however a PF cannot be applied directly due to the large dimensionality of the state space. Instead, elements of the state vector are grouped into zones as

$$\begin{aligned} \Theta_g &= \left[\vec{\theta}_1, \ldots, \vec{\theta}_N\right] \quad \text{where} \\ \vec{\theta}_i &= [\vec{\mu}_i, \vec{w}_i, \sigma_i] \qquad \text{for } i \in [1, N] \end{aligned} \tag{173}$$

and Eq. (166) is minimized with respect to a single zone $\vec{\theta}_i$ at a time:

$$\vec{\theta}_i^* = \underset{\vec{\theta}_i}{\operatorname{argmin}} \ R\Big(\vec{\theta}_i \mid \Theta_{\neg i}, \hat{\Theta}_s, \Theta_l\Big) \tag{174}$$

where $\Theta_g = \left[\Theta_{\neg i}, \vec{\theta}_i\right]$. Then, the process described in Section 4.3.2 is applied to find the optimal Θ_g.

5.3.3 Gradient Descent Optimization of $\mathcal{L}_l$

Due to the low number of bases entering the parametrization of $\mathcal{L}_g$ in Eq. (162) (empirically determined, $N \approx 10$ is a good choice), gross misalignments are captured, but fine misregistrations remain. To eliminate these errors, a larger number of bases is required. However, a stochastic approach becomes too computationally intensive and does not provide significant benefit because the remaining deformations are local. Hence, in the last level of registration, we use a larger number of Gaussian bases in the representation of $\mathcal{L}_l$ and search for the best parameters with a gradient descent method. The bases are distributed according to a uniform grid covering the entire domain, a common standard deviation β is set by the user, and the weights $\vec{w}_j$ are optimised. Computation of the gradient with respect to the basis weights is:

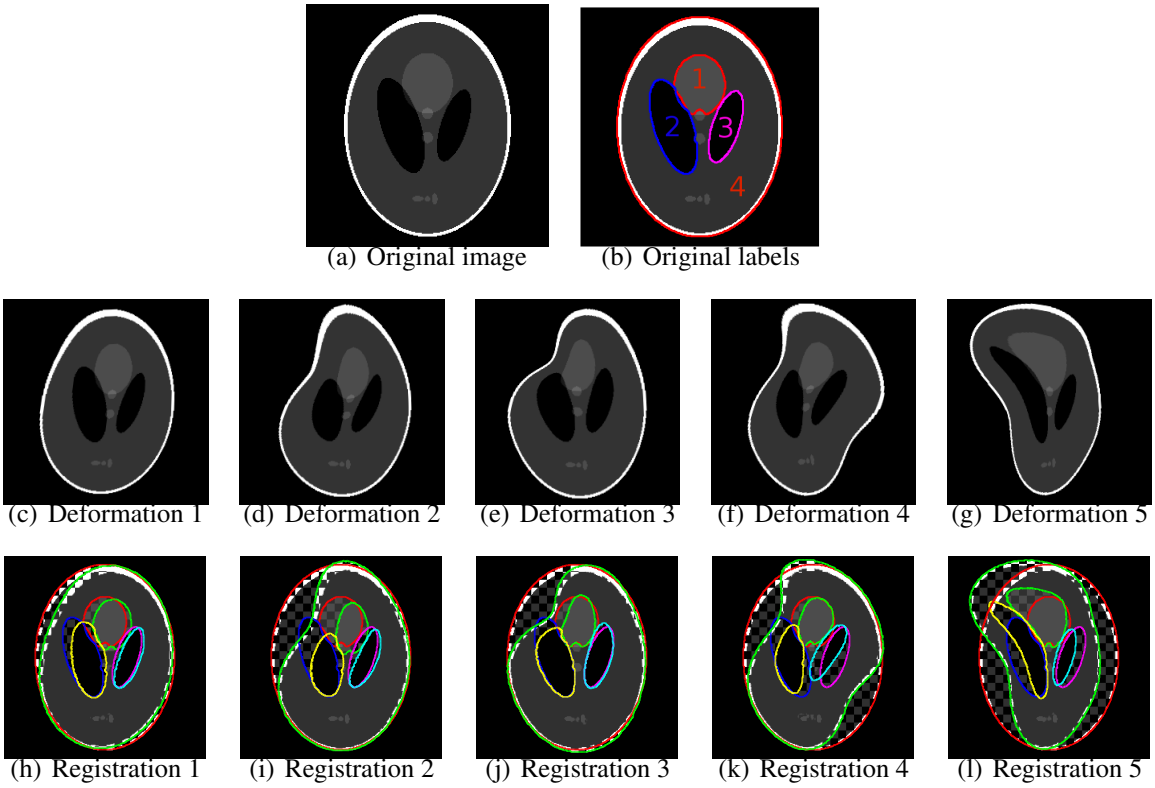
$$\frac{\partial E(\Theta_l)}{\partial \vec{w}_j} = \int_\Omega -2\Big(I_M(\vec{x}) - I_D(\mathcal{L}(\vec{x}))\Big)\Big(\nabla I_D(\mathcal{L})\Big)^T \frac{\partial \mathcal{L}_l}{\partial \vec{w}_j}\, d\Omega + \lambda \frac{\partial E_r(\Theta_l)}{\partial \vec{w}_j}\,. \qquad (175)$$

The injectivity constraint is enforced by performing a re-gridding of the domain if $d_J(\cdot)$ falls below a chosen threshold. At each time a re-gridding is performed, the image is warped forward according to the current deformation field and the algorithm re-starts. The final deformation is a composition of all intermediate maps and is guaranteed to preserve the topology of the original space.

5.4 Results

In this section, the proposed *SIR* approach is quantitatively evaluated on synthetic data against two state of the art registration algorithms: *SyN* [6] and *NiftyReg* [81]. Testing data consists of five synthetic examples, depicted in Fig. 35. These examples were created by applying a series of warps of increasing severity to the original phantom in Fig. 35(a). We selected four structures in the phantom, which are labeled and outlined in Fig. 35(b); these structures are the ground truth label maps. Then, for each of the examples, a set of corresponding label maps was made by warping the ground truth with the same deformation used to create the testing image pairs. The resulting synthetic tests are shown as five columns in Fig. 35; the top row contains the deformed images and the bottom row uses a checkerboard pattern to alternatively display the target image and moving image and overlays the ground truth labels and the moving labels on top of the images.

In Fig. 36, Fig. 37 and Fig. 38 the registration results for *SyN*, *NiftyReg*, and SIR, respectively, are illustrated. Parameters for *SyN* and *NiftyReg* were set to the best of our abilities, and a hierarchical approach was used for each algorithm starting from a coarse resolution, computing the coarse warp, and increasing the resolution at the next stage. The proposed SIR approach contains three user defined parameters: the number of basis functions N used for $\mathcal{L}_g$ as well as the spacing Δ and standard deviation β of the bases



(a) Original image (b) Original labels

(c) Deformation 1 (d) Deformation 2 (e) Deformation 3 (f) Deformation 4 (g) Deformation 5

(h) Registration 1 (i) Registration 2 (j) Registration 3 (k) Registration 4 (l) Registration 5

Figure 35: A series of synthetic images used for testing are shown. The starting, target image is shown in Fig. 35(a); corresponding labels numbered 1-4 are seen in Fig. 35(b). A thin plate spline was used to deform this image and to create the warped images in Fig. 35(c) - 35(g). The severity of the warp increases from left to right. In Fig. 35(h) - Fig. 35(l), the target (image and labels) and corresponding deformed image from the upper row are overlayed; additionally, ground truth and warped labels for four structures in the phantom are shown. These labels will be used for evaluating registration results.

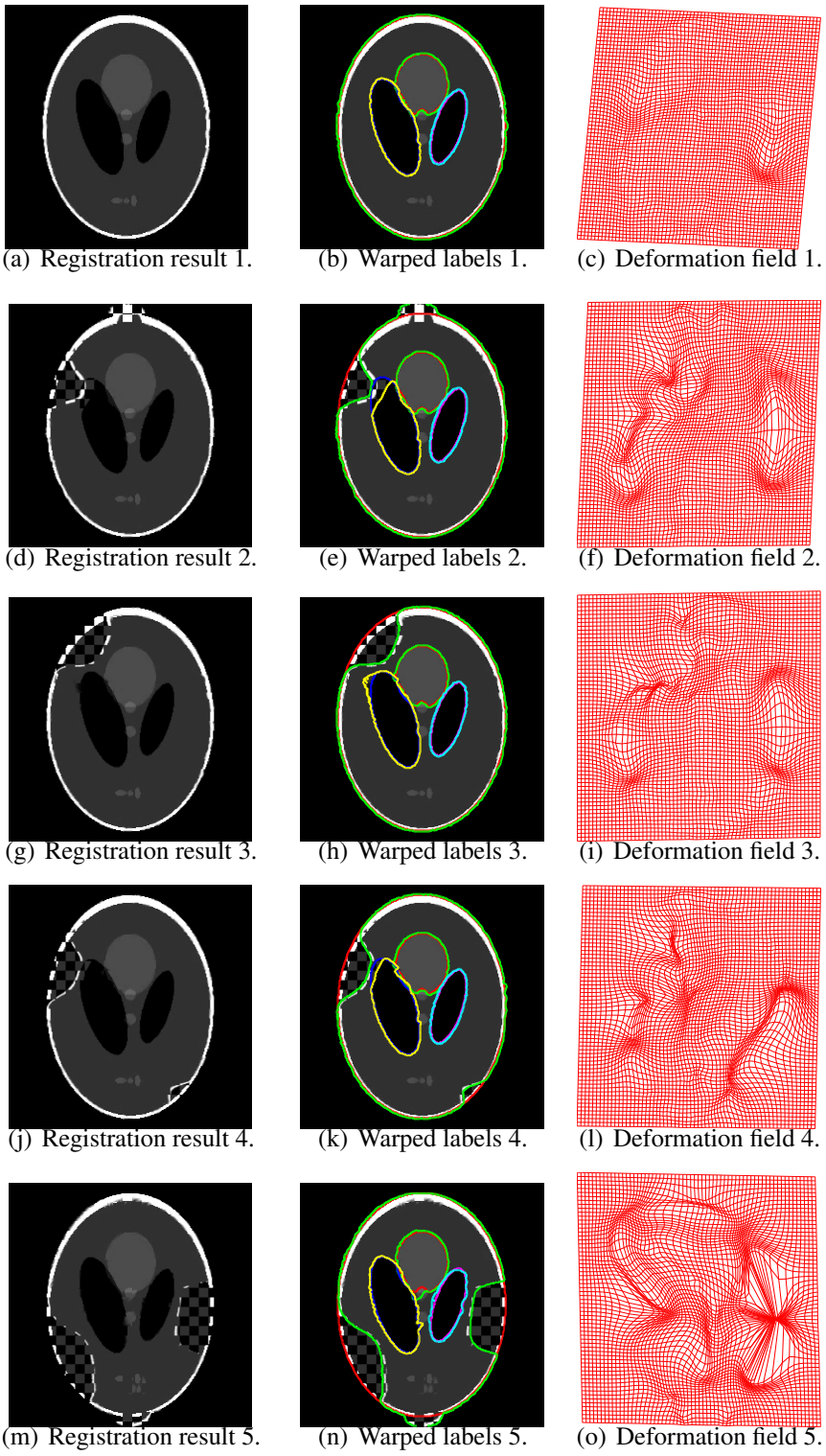
	SyN	NiftyReg	SIR
Registration 1	0.966 0.971 0.955 0.991	0.953 0.960 0.935 0.989	0.967 0.967 0.954 0.991
Registration 2	0.969 0.916 0.947 0.970	0.886 0.865 0.933 0.956	0.965 0.966 0.952 0.991
Registration 3	0.969 0.948 0.959 0.967	0.936 0.954 0.935 0.970	0.961 0.969 0.953 0.992
Registration 4	0.958 0.946 0.957 0.969	0.957 0.953 0.951 0.966	0.965 0.965 0.952 0.991
Registration 5	0.948 0.961 0.941 0.914	0.869 0.917 0.930 0.966	0.961 0.960 0.949 0.992

Table 2: This table evaluates quantitatively the registration accuracy for the four methods *SyN*, *NiftyReg*, and *SIR*. Ground truth label maps are available for four structures in the phantom, as seen in Fig. 35(b). The registration fields computed in the third column of Fig. 36 (*SyN*) , Fig. 37 (*NiftyReg*), and Fig. 38 (*SIR*) are applied to the deformed label maps. In each table cell, Dice coefficients are reported for labels 1-4, respectively, for a particular method/registration test pair. An exact match between ground truth and the registered label maps would produce a Dice coefficient of 1.

composing $\mathcal{L}_l$. These parameters were set to $N = 14$, $\Delta = \frac{1}{10}$, $\beta = \frac{1}{10}$, assuming that the image is contained inside of a square of size one.

Dice coefficients can be used to measure the degree of overlap between a ground truth label and the corresponding registered label; these measures are reported in Table 2. Registered labels were computed by applying the deformation found during image registration to the deformed labels. In each cell of the table, there are four values: Dice coefficients for labels 1-4 from Fig. 35(b). Thus, the degree of label overlap after registration using each of the three algorithms is evaluated. It is clear that SIR performs better for larger deformations, as seen by the consistently high Dice coefficients, especially in Registrations 2-5. In all experiments, all four structures are registered virtually perfectly using the SIR method.

SIR was also tested on a 3D MRI volume of the same patient in two different conditions: the images are of the buttocks, one when the person is sitting on a hard surface and the other when she is suspended and there is no contact of the tissue with the surface. As evidenced by the left column of Fig. 39, the soft tissues in the region experience significant shifting and compression. As a result, the initial misalignment is large. SIR corrects even for these large displacements and the aligned images can be seen in the right column of Fig. 39(f).



(a) Registration result 1. (b) Warped labels 1. (c) Deformation field 1.

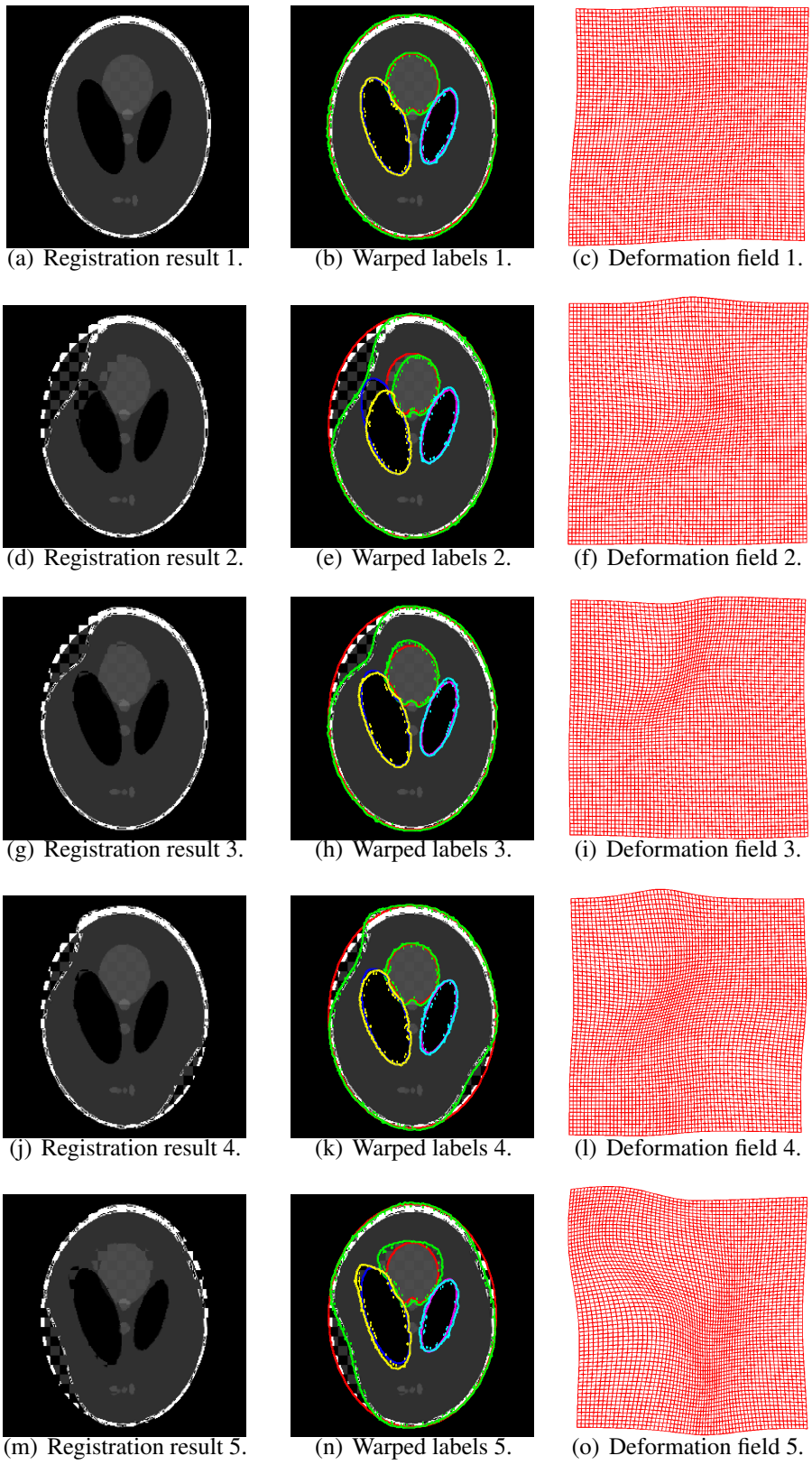
(d) Registration result 2. (e) Warped labels 2. (f) Deformation field 2.

(g) Registration result 3. (h) Warped labels 3. (i) Deformation field 3.

(j) Registration result 4. (k) Warped labels 4. (l) Deformation field 4.

(m) Registration result 5. (n) Warped labels 5. (o) Deformation field 5.

Figure 36: Registration results using the *SyN* [6] algorithm on the synthetic image pairs from Fig. 35. The first column shows the target image with the non-rigidly registered image overlayed; the visualization uses a checkerboard pattern to alternate between displaying the target or the registered image regions. The second column overlays the label maps warped using the computed deformation along with the ground truth labels, and the third column displays the deformation field that was found.



(a) Registration result 1. (b) Warped labels 1. (c) Deformation field 1.

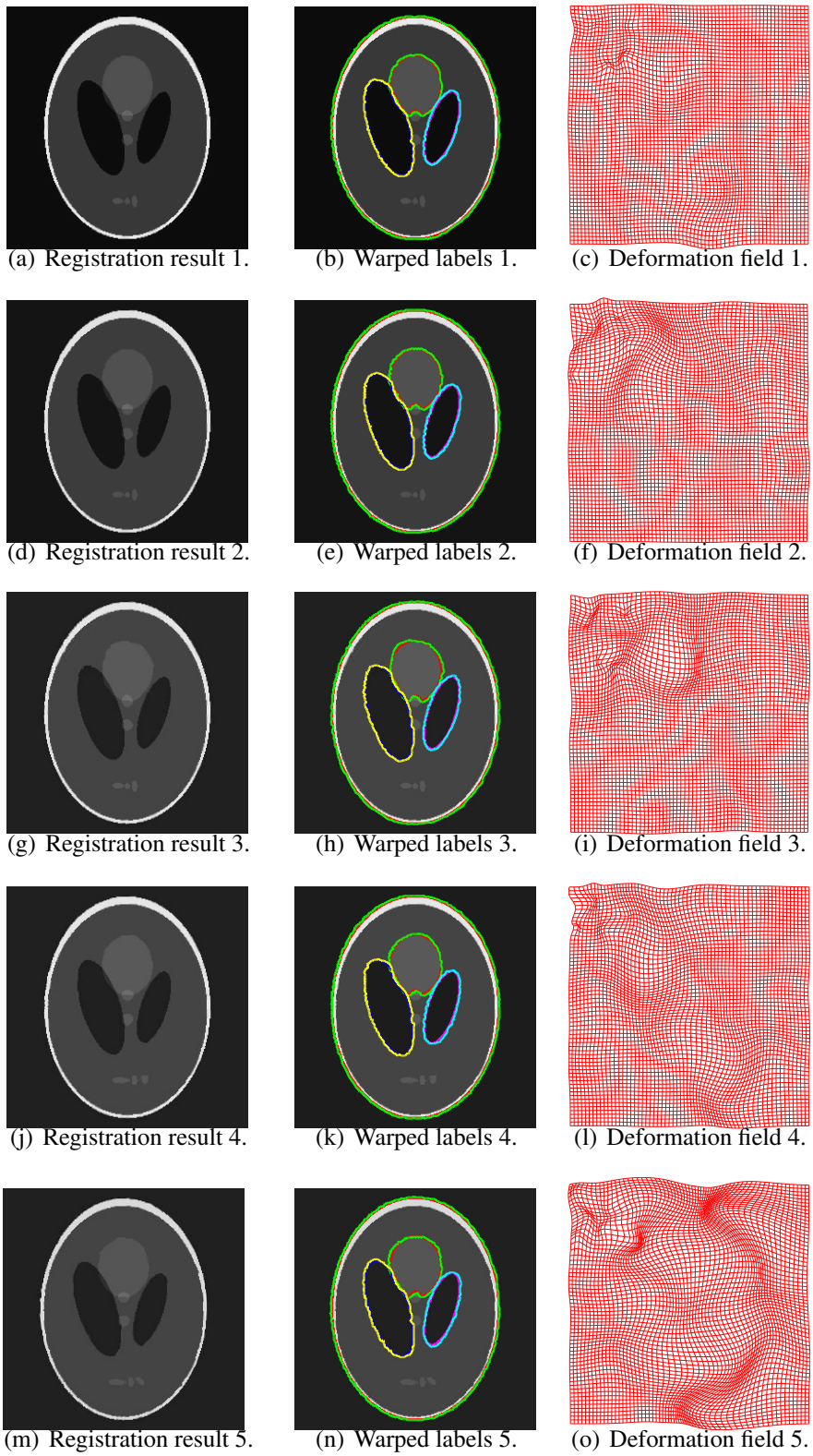
(d) Registration result 2. (e) Warped labels 2. (f) Deformation field 2.

(g) Registration result 3. (h) Warped labels 3. (i) Deformation field 3.

(j) Registration result 4. (k) Warped labels 4. (l) Deformation field 4.

(m) Registration result 5. (n) Warped labels 5. (o) Deformation field 5.

Figure 37: Registration results for *NiftyReg* [81]. The first column overlays the target and registered image, the second adds the registered label maps, and the third column contains the computed deformation (same visualizations as in Fig. 36 but results for *NiftyReg*) .



(a) Registration result 1. (b) Warped labels 1. (c) Deformation field 1.

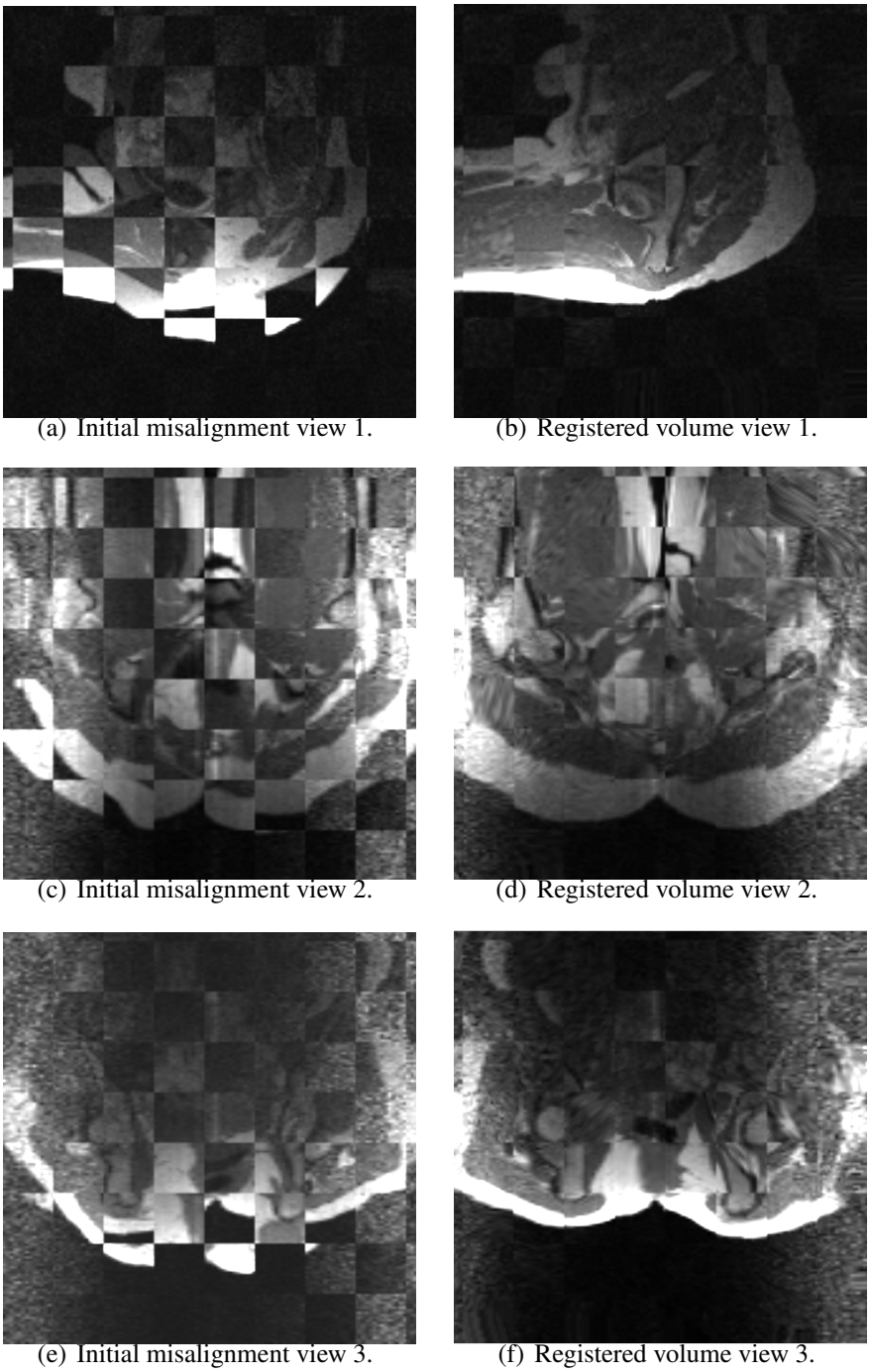
(d) Registration result 2. (e) Warped labels 2. (f) Deformation field 2.

(g) Registration result 3. (h) Warped labels 3. (i) Deformation field 3.

(j) Registration result 4. (k) Warped labels 4. (l) Deformation field 4.

(m) Registration result 5. (n) Warped labels 5. (o) Deformation field 5.

Figure 38: Registration results for the *SIR* algorithm proposed in this note. The first column overlays the target and registered image, the second adds the registered label maps, and the third column contains the computed deformation field (same visualizations as Fig. 36 and Fig. 37 to display *SIR* results).



(a) Initial misalignment view 1. (b) Registered volume view 1.

(c) Initial misalignment view 2. (d) Registered volume view 2.

(e) Initial misalignment view 3. (f) Registered volume view 3.

Figure 39: Registration results for two 3D MRI volumes with the proposed *SIR* algorithm. Three representative slices are shown before registration in the left column with the target and moving images slices mixed according to a checkerboard pattern. The same slices are shown after alignment in the right column.

5.5 Chapter Conclusion

This chapter presents a stochastic image registration methodology that is robust to large deformations. The deformation field $\mathcal{L}$ is parametrized as an additive composition of a similarity transformation, a global non-rigid transformation, and a local non-rigid deformation through a GRBF network. Important constraints on the transformation $\mathcal{L}$ are imposed to ensure physically realistic deformations. The proposed *SIR* approach is shown to be a powerful tool to capture large deformations.

Several improvements to the approach are envisioned. First, by replacing the symmetric Gaussian basis functions in Eq. (85) with full covariance matrices, the number of basis functions required to represent a deformation of a particular complexity will be reduced and potentially likewise for the optimization space. Also, image registration with constraints is envisioned in which the user can guide the registration by denoting stationary areas in the image domain while automatic registration is performed in the remaining regions.

CHAPTER VI

CONCLUDING REMARKS AND FUTURE RESEARCH

This thesis analyzes the failure mechanisms for commonly used segmentation and registration methods and tackles these problems from different ways, depending on the source of difficulty. The global idea is to include the user in the segmentation/registration framework and allow him to interact with a semi-automatic algorithm to complete the computer vision task. By allowing this interaction, the user lends his experience and *expert knowledge* to guide the semi-automatic method while leveraging the speed of the autonomous algorithm. Then, the challenge lies in efficiently taking advantage of the user input; the approach for doing so depends on the difficulty of problem and the prior knowledge available.

In the first chapter, a segmentation method is proposed for the case in which no training data or shape model is available and existing intensity based segmentation energies are not discriminative for the target structure. Previously, all methods would have failed and the user would have been left to do the segmentation completely by hand, a tedious, slow process. The HSC approach described in Chapter 2 solves this problem by periodically displaying the segmentation result and allowing the user to influence the speed of the segmenting curve with his input. As a result, label maps of manual accuracy are obtained in a fraction of the time.

For medical imaging segmentation, shape models have proven to be a powerful source of prior knowledge. In a number of applications, including adaptive radio-therapy, a sequence of segmentations is required. Clearly, with each completed segmentation, additional knowledge is gained about the patient's anatomy, and completed label maps should serve as landmarks. Chapter 3 proposes to use not only information about the shape of a single structure but also relative shapes and locations of pairs of structures. These pairwise

models allow the algorithm to predict the locations of the next structures in a sequence, and with each newly segmented object, positions of future segmentation are refined.

Depending on the complexity of the image, existing registration algorithms have varying success rate. The registration task can be greatly simplified for an automatic approach with a simple operation from the user. For instance, a binary bone mask can be generated, in CT imagery with just a threshold. Then, registration can be performed on a point cloud, which is a sparse representation of the original image. Chapter 4 proposes a framework for doing constrained point set registration. This framework combines user input and known physical constraints to regularize the problem. A novel stochastic optimization method allows the algorithm to escape from local minima and makes it robust to parameter settings.

Even if a registration cost does have the global optimum corresponding to the desired alignment, local optima can be a reason for poor results. In particular, large misalignments result in state-of-the-art registration approaches becoming trapped in local minima during optimization. Hence, in Chapter 5, we propose a hierarchical method for capturing large deformations. The approach consists of stochastic registration for rough alignment and then a gradient descent with a larger number of bases for capturing small, remaining errors. The results show excellent performance on synthetic and real, MRI imagery.

The research described in this dissertation investigates a number of relevant, unsolved problems in image segmentation and registration and provides solutions to a subset of them. Admittedly, each chapter can be expanded, which is left for future research. Motivated by the approach of Chapter 2 for segmentation, an interesting research direction is to develop an interactive registration technique that is able to attain ground truth results for complex images. It is of interest to evaluate different methods for fusing label maps to form the predicted segmentation in Chapter 3. Exploring options for efficiently initializing and adaptively changing the dynamics for different registration cases would be fruitful for improving the convergence rate of the registration method described in Chapter 4. Furthermore, significant room is left for proposing useful constraints in the point set registration framework.

Similarly, adding constraints for image registration, in Chapter 5, is expected to make the method robust in a wider spectrum of applications.

REFERENCES

[1] A. TOP, G. H. and ABUGHARBIEH, R., "Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3d image segmentation," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the*, 2010.

[2] AIT-AOUDIA, S. and MAHIOU, R., "Medical image registration by simulated annealing and genetic algorithms," in *International Conference on Geometric Modeling and Imaging (GMAI), Proceedings of the*, pp. 145 –148, July 2007.

[3] ALJABAR, P., HECKEMANN, R., HAMMERS, A., HAJNAL, J., and RUECKERT, D. *NeuroImage*, vol. 46, no. 3, pp. 726 – 738, 2009.

[4] ANDRIEU, C., DE FREITAS, N., DOUCET, A., and JORDAN, M., "An Introduction to MCMC for Machine Learning," *Machine Learning*, vol. 50, no. 1, pp. 5 – 43, 2003.

[5] ARCE-SANTANA, E., CAMPOS-DELGADO, D., and ALBA, A., "A non-rigid multimodal image registration method based on particle filter and optical flow," in *International Conference on Advances in Visual Computing (ISVC), Proceedings of the*, (Berlin, Heidelberg), pp. 35 – 44, Springer-Verlag, 2010.

[6] AVANTS, B., DUDA, J., KIM, J., ZHANG, H., PLUTA, J., GEE, J., and WHYTE, J., "Multivariate analysis of structural and diffusion imaging in traumatic brain injury," *Academic Radiology*, vol. 15, no. 11, pp. 1360 – 1375, 2008.

[7] BACH, F. and JORDAN, M., "A Probabilistic Interpretation of Canonical Correlation Analysis," Tech. Rep. 688, Department of Statistics Univeristy of California, Berkeley, Apr. 2005.

[8] BARRETT, W. and MORTENSEN, E., "Interactive live-wire boundary extraction," *Medical Image Analysis*, vol. 1, pp. 331 – 341, 1997.

[9] BESL, P. and MCKAY, H., "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 14, pp. 239 – 256, Feb. 1992.

[10] BING, J. and VEMURI, B., "Robust point set registration using gaussian mixture models," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 33, pp. 1633 – 1645, Aug. 2011.

[11] BOOKSTEIN, F. L., "Principal warps: Thin-plate splines and the decomposition of deformations," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 11, no. 6, pp. 567 – 585, 1989.

[12] BOYKOV, Y. and JOLLY, M., "Interactive graph cuts for optimal boundary region segmentation of objects in n-d images," in *International Conference on Computer Vision (ICCV), Proceedings of the*, (Vancouver, B.C., Canada), July 2001.

[13] BOYKOV, Y. and JOLLY, M., "Interactive graph cuts for optimal boundary region segmentation of objects in n-d images," in *International Conference on Computer Vision (ICCV), Proceedings of the*, (Vancouver, B.C., Canada), July 2001.

[14] BOYKOV, Y., VEKSLER, O., and ZABIH, R., "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, Nov. 2001.

[15] BROWN, M. and LOWE, D., "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision (IJCV)*, vol. 74, no. 1, pp. 59 – 73, 2007.

[16] CARREIRA-PERPINAN, M., "Mode-finding for mixture of gaussian distributions," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 22, no. 11, pp. 1318 – 1323, 2000.

[17] CASELLES, V., KIMMEL, R., and SAPIRO, G., "Geodesic active contours," *International Journal of Computer Vision (IJCV)*, vol. 22, no. 1, pp. 61 – 79, 1997.

[18] CATES, J., LEFOHN, A., and WHITAKER, R., "Gist: An interactive, gpu-based level set segmentation tool for 3d medical images," *Medical Image Analysis*, vol. 8, pp. 217 – 231, 2004.

[19] CHAN, T. and VESE, L., "Active contours without edges," *Image Processing (TIP), IEEE Transactions on*, vol. 10, no. 2, pp. 266 – 277, 2001.

[20] CHEN, Z., "Bayesian filtering: From kalman filters to particle filters, and beyond," tech. rep., McMaster University, 2003.

[21] CHRISTENSEN, G., RABBITT, R., and MILLER, M., "Deformable templates using large deformation kinematics," *Image Processing (TIP), IEEE Transactions on*, vol. 5, no. 10, pp. 1435 – 1447, 1996.

[22] CHUI, H. and RANGARAJAN, A., "A feature registration framework using mixture models," in *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA), Proceedings of the*, pp. 190 – 197, 2000.

[23] CHUI, H. and RANGARAJAN, A., "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114 – 141, 2003.

[24] COLLIGNON, A., MAES, F., DELAERE, D., VANDERMEULEN, D., SUETENS, P., and MARCHAL, G., "Automated multimodality image registration based on information theory," in *International Conference on Information Processing in Medical Imaging (IPMI), Proceedings of the*, vol. 3, pp. 263 – 274, Kluwer Academic Publishers, 1995.

[25] DAMBREVILLE, S., RATHI, Y., and TANNENBAUM, A., "A shape-based approach to robust image segmentation using kernel pca," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Proceedings of the*, pp. 977 – 984, 2006.

[26] DEL MORAL, P., DOUCET, A., and JASRA, A., "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 3, pp. 411 – 436, 2006.

[27] DEL MORAL, P., DOUCET, A., and JASRA, A., "Sequential monte carlo for bayesian computation," vol. 8, p. 34, Oxford University Press, USA, 2007.

[28] D'HAESE, P., DUAY, V., MERCHANT, T., MACQ, B., and DAWANT, B., "Atlas-based segmentation of the brain for 3-dimensional treatment planning in children with infratentorial ependymoma," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the* (ELLIS, R. and PETERS, T., eds.), vol. 2879 of *Lecture Notes in Computer Science*, pp. 627 – 634, Springer Berlin / Heidelberg, 2003.

[29] DOUCET, A., GODSILL, S., and ANDRIEU, C., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197 – 208, 2000.

[30] DOUCET, A. and JOHANSEN, A., "A tutorial on particle filtering and smoothing: Fifteen years later," *The Oxford Handbook of Nonlinear Filtering*, no. Dec., 2011.

[31] EHLERS, M. and FOGEL, D., "High-precision geometric correction of airborne remote sensing revisited: the multiquadric method," in *SPIE Image and Signal Processing for Remote Sensing, Proceedings of the*, pp. 814 – 824, Sept. 1994.

[32] GAO, Y., TANNENBAUM, A., and KIKINIS, R., "Simultaneous multi-object segmentation using local robust statistics and contour interaction," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) Workshop on Medical Computer Vision, Proceedings of the*, 2010.

[33] GILKS, W. and BERZUINI, C., "Following a moving target?monte carlo inference for dynamic bayesian models," *Journal of the Royal Statistical Society: Series B*, vol. 63, no. 1, pp. 127 – 146, 2001.

[34] GORDON, N., SALMOND, D., and SMITH, A., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *Radar and Signal Processing, IEE Proceedings F*, vol. 140, pp. 107 – 113, 1993.

[35] GUSTAFSSON, F., GUNNARSSON, F., BERGMAN, N., FORSSELL, U., JANSSON, J., KARLSSON, R., and NORDLUND, P., "Particle filters for positioning, navigation, and tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 425 – 437, 2002.

[36] G.WAHBA, ed., *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.

[37] HARRIS, C. and STEPHENS, M., "A Combined Corner and Edge Detection," in *Alvey Vision Conference, Proceedings of the*, pp. 147 – 151, 1988.

[38] HECKER, F., KONRAD, O., HORST, K., and PEITGEN, H., "Interactive 3d medical image segmentation with energy-minimizing implicit functions," *Journal of Computers and Graphics*, vol. 35, no. 2, pp. 275 – 287, 2011.

[39] HOLDEN, M., "A Review of Geometric Transformations for Nonrigid Body Registration," *Medical Imaging (TMI), IEEE Transactions on*, vol. 27, no. 1, pp. 111 – 128, 2007.

[40] JACKSON, E., ed., *A User's Guide to Principle Components*. New York, NY: John Wiley & Sons, Inc., 1991.

[41] JENKINSON, M. and SMITH, S., "A global optimisation method for robust affine registration of brain images.," *Medical Image Analysis*, vol. 5, pp. 143 – 156, June 2001.

[42] JI, C., ZHANG, Y., TONG, M., and YANG, S., "Particle filter with swarm move for optimization," in *International conference on Parallel Problem Solving from Nature (PPSN), Proceedings of the*, (Berlin, Heidelberg), pp. 909 – 918, Springer-Verlag, 2008.

[43] JIAN, B. and VEMURI, B., "A robust algorithm for point set registration using mixture of gaussians," vol. 2, (Los Alamitos, CA, USA), pp. 1246 – 1251, IEEE Computer Society, 2005.

[44] JUNCK, L., MOEN, J., HUTCHINS, G., BROWN, M., and KUHL, D., "Correlation methods for the centering, rotation, and alignment of functional brain images," *The Journal of Nuclear Medicine (JNM)*, vol. 31, pp. 1220 – 1276, 1990.

[45] JUNG, H., LEE, K., and LEE, S., "Window annealing over square lattice markov random field," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, European Conference on Computer Vision (ECCV), Proceedings of the, (Berlin, Heidelberg), pp. 307 – 320, Springer-Verlag, 2008.

[46] KARASEV, P., KOLESOV, I., CHUDY, K., MULLER, G., XEROGEANES, J., and TANNENBAUM, A., "Interactive mri segmentation with controlled active vision," in *IEEE Conference on Decision and Control, Proceedings of the*, IEEE, Dec. 2011.

[47] KASETKASEM, T., HOMSUP, N., and MEETIT, D., "An image registration algorithm using a particle filters," in *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Proceedings of the*, vol. 02, pp. 1120 – 1123, May 2009.

[48] KASS, M., WITKIN, A., and TERZOPOULOS, D., "Snakes: Active contour models," *International Journal of Computer Vision (IJCV)*, vol. 1, no. 4, pp. 321 – 331, 1988.

[49] KICHENESAMY, S., KUMAR, A., OLVER, P., TANNENBAUM, A., and YEZZI, A., "Conformal curvature flows: From phase transitions to active contours," *Archive for Rational Mechanics and Analysis*, vol. 134, no. 1, pp. 275 – 301, 1996.

[50] KIRKPATRICK, S., GELATT, C. D., and VECCHI, M. P., "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671 – 680, 1983.

[51] KOLESOV, I., KARASEV, P., MULLER, G., CHUDY, K., XEROGEANES, J., and TANNENBAUM, A., "Human supervisory control framework for interactive medical image segmentation," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) Computational Biomechanics for Medicine Workshop, Proceeding of the*, 2011.

[52] LANKTON, S. and TANNENBAUM, A., "Localizing region-based active contours," *Image Processing (TIP), IEEE Transactions on*, vol. 17, no. 11, pp. 2029 – 2039, 2008.

[53] LEVENTON, M., GRIMSON, E., and FAUGERAS, O., "Statistical shape influence in geodesic active contours," vol. 1, pp. 316 – 323, 2000.

[54] LEVENTON, M. E., GRIMSON, W. E. L., and FAUGERAS, O., "Statistical shape influence in geodesic active contours," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Proceedings of the*, vol. 1, pp. 316 – 323, 2000.

[55] LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., and FULK, D., "The digital michelangelo project: 3d scanning of large statues," International Conference and Exhibition of Computer Graphics and Interactive Techniques SIGGRAPH, Proceeding of the, pp. 131 – 144, 2000.

[56] LOCATELLI, M., "Simulated annealing algorithms for continuous global optimization: convergence conditions," *Journal of Optimization Theory and Applications*, vol. 104, pp. 121 – 133, Jan. 2000.

[57] LOECKX, D., MAES, F., VANDERMEULEN, D., and SUETENS, P., "Nonrigid Image Registration Using Free-Form Deformations with a Local Rigidity Constraint," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the*, vol. 3216 of *Lecture Notes in Computer Science*, ch. 78, pp. 639 – 646, Springer Berlin / Heidelberg, 2004.

[58] LOU, Y., JIA, X., GU, X., and TANNENBAUM, A., "A gpu-based implementation of multimodal deformable image registration based on mutual information or bhattacharyya distance," *The Insight Journal*, May 2011.

[59] MA, B. and ELLIS, Y., "Surface-based registration with a particle filter," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the*, pp. 566 – 573, 2004.

[60] MAHFOUZ, M., HOFF, W., KOMISTEK, R., and DENNIS, D., "A robust method for registration of three-dimensional knee implant models to two-dimensional flouroscopy images," *Medical Imaging (TMI), IEEE Transactions on*, vol. 22, no. 12, pp. 1561 – 1574, 2003.

[61] MEISTERS, G. and OLECH, C., "Locally one-to-one mappings and a classical theorem on schlicht functions.," *Duke Mathematics Journal*, vol. 30, pp. 63 – 80, 1963.

[62] MEYER, C., BOES, J., KIM, B., BLAND, P., ZASADNY, K., KISON, P., KORAL, K., FREY, K., and WAHL, R., "Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin-plate spline warped geometric deformations," *Medical Image Analysis*, vol. 1, pp. 195 – 206, Apr. 1997.

[63] MICHAILOVICH, O., RATHI, Y., and TANNENBAUM, A., "Image segmentation using active contours driven by the bhattacharyya gradient flow," *Image Processing (TIP), IEEE Transactions on*, vol. 16, no. 11, pp. 2787 – 2801, 2007.

[64] MODERSITZKI, J., *Numerical Methods for Image Registration*. Oxford University Press, 2004.

[65] MUSSO, C., OUDJANE, N., and LEGLAND, F., "Improving regularised particle filters," 2001.

[66] MYRONENKO, A. and SONG, X., "Point Set Registration: Coherent Point Drift," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 32, no. 12, pp. 2262 – 2275, 2010.

[67] OSHER, S. and FEDKIW, R., *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences, Springer, 2002.

[68] OSHER, S. and SETHIAN, J., "Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation," *Journal of Computational Physics*, vol. 79, pp. 12 – 49, 1988.

[69] OTSU, N., "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics (SMC), IEEE Transactions on*, vol. 9, no. 1, pp. 62 – 66, 1979.

[70] PARK, J. and SANDBERG, I. W., "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246 – 257, June 1991.

[71] PLUIM, J. P. W., MAINTZ, J. B. A., and VIERGEVER, M. A., "Mutual-information-based registration of medical images: a survey," *Medical Imaging (TMI), IEEE Transactions on*, vol. 22, no. 8, pp. 986 – 1004, 2003.

[72] PRATT, W., *Digital Image Processing*. John Wiley & Sons, Inc., 2007.

[73] RANGARAJAN, A., CHUI, H., and BOOKSTEIN, F., "The softassign procrustes matching algorithm," in *International Conference on Information Processing in Medical Imaging, Proceedings of*, pp. 29 – 42, Springer, 1997.

[74] RAVIV, T., VAN-LEEMPUT, K., WELLS III, W., and GOLLAND, P., "Joint segmentation of image ensembles via latent atlases," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the*, (London, UK), pp. 272 – 280, Sept. 2009.

[75] REITER, M., DONNER, R., LANGS, G., and BISCHOF, H., "Estimation of face depth maps from color texture using canonical correlation analysis," in *Computer Vision Winter Workshop, Proceedings of the*, (Telc, Czech Republic), Feb. 2006.

[76] RISTIC, B., ARULAMPALAM, S., and GORDON, N., *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.

[77] ROBERT, C. and CASELLA, G., *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[78] ROHR, K., STIEHL, H., SPRENGEL, R., BUZUG, T., WEESE, J., and KUHN, M. H., "Landmark-based elastic registration using approximating thin-plate splines.," *Medical Imaging (TMI), IEEE Transactions on*, pp. 526 – 534, 2001.

[79] ROUSSON, M. and PARAGIOS, N., "Shape priors for level set representations," in *European Conference on Computer Vision (ECCV), Proceedings of the*, pp. 78 – 92, Springer, 2002.

[80] RUECKERT, D., ALJABAR, P., HECKEMANN, R., HAJNAL, J., and HAMMERS, A., "Diffeomorphic registration using b-splines," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the*, pp. 702 – 709, 2006.

[81] RUECKERT, D., SONODA, L., HAYES, C., HILL, D., LEACH, M., and HAWKES, D., "Nonrigid registration using free-form deformations: Application to breast mr images," *Medical Imaging (TMI), IEEE Transactions on*, vol. 18, pp. 712 – 721, 1999.

[82] SABUNCU, M., YEO, B., VAN LEEMPUT, K., FISCHL, B., and GOLLAND, P., "A generative model for image segmentation based on label fusion," *Medical Imaging (TMI), IEEE Transactions on*, vol. 29, pp. 1714 – 1729, Oct. 2010.

[83] SANDHU, R., DAMBREVILLE, S., and TANNENBAUM, A., "Point set registration via particle filtering and stochastic dynamics," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 32, pp. 1459 – 1473, 2010.

[84] SCHARSTEIN, D. and SZELISKI, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision (IJCV)*, vol. 47, pp. 7 – 42, 2001.

[85] SETHIAN, J., ed., *Level Set Methods and Fast Marching Methods*. New York, NY: Springer, 1999.

[86] SHERIDAN, T., ed., *Telerobotics, automation, and human supervisory control*. Cambridge, MA: MIT Press, 1992.

[87] STUDHOLME, C., HILL, D. L. G., and HAWKES, D. J., "Multiresolution voxel similarity measures for MR-PET registration," in *International Conference on Information Processing in Medical Imaging (IPMI), Proceedings of the*, pp. 287 – 298, 1995.

[88] STUDHOLME, C., HILL, D., and HAWKES, D., "Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures," *Medical Physics*, vol. 24, no. 1, pp. 25 – 35, 1997.

[89] STUDHOLME, C., HILL, D., and HAWKES, D., "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition*, vol. 32, pp. 71 – 86, Jan. 1999.

[90] THIRION, J., "Image matching as a diffusion process: an analogy with Maxwell's demons," *Medical Image Analysis*, vol. 2, no. 3, pp. 243 – 260, 1998.

[91] TIPPING, M. and BISHOP, C., "Probabilistic principal component analysis," *Journal of the Royal Statistical Society*, vol. 61, no. 3, pp. 611 – 622, 1999.

[92] TOGA, A., *Brain Warping*. Academic Press, 1999.

[93] TSAI, A., YEZZI, A. J., WELLS, W., TEMPANY, C., TUCKER, D., FAN, A., GRIMSON, W. E., and WILLSKY, A., "A shape-based approach to the segmentation of medical imagery usinglevel sets," *Medical Imaging (TMI), IEEE Transactions on*, vol. 22, no. 2, pp. 137 – 154, 2003.

[94] TSIN, Y. and KANADE, T., "A correlation-based approach to robust point set registration," in *European Conference on Computer Vision (ECCV), Proceedings of the*, pp. 558 – 569, 2004.

[95] VAN DEN ELSEN, P., POL, E., and VIERGEVER, M., "Medical image matching-a review with classification," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 12, pp. 26 – 39, Mar. 1993.

[96] VANDERBILT, D. and LOUIE, S., "A monte carlo simulated annealing approach to optimization over continous variables," *Journal of Computational Physics*, vol. 56, no. 2, pp. 259 – 271, 1984.

[97] VERCAUTEREN, T., PENNEC, X., PERCHANT, A., and AYACHE, N., "Diffeomorphic demons: Efficient non - parametric image registration.," *NeuroImage*, vol. 45, no. 1, pp. 61 – 72, 2008.

[98] VERCAUTEREN, T., PENNEC, X., PERCHANT, A., and AYACHE, N., "Diffeomorphic demons: Efficient non-parametric image registration," *NeuroImage*, vol. 45, no. 1, pp. 61 – 72, 2009. Mathematics in Brain Imaging.

[99] VIOLA, P. and WELLS III, W., "Alignment by maximization of mutual information," in *International Conference on Computer Vision (ICCV), Proceedings of the*, 1995.

[100] WANG, F., VEMURI, B., RANGARAJAN, A., and EISENSCHENK, S., "Simultaneous nonrigid registration of multiple point sets and atlas construction," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 30, no. 11, pp. 2011 – 2022, 2008.

[101] WANG, H., WOOK SUH, J., DAS, S., PLUTA, J., ALTINAY, M., and YUSHKEVICH, P., "Regression-based label fusion for multi-atlas segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Proceedings of the*, pp. 1113 – 1120, June 2011.

[102] WANG, J. and JIANG, T., "Nonrigid Registration of Brain MRI Using NURBS," *Pattern Recognition Letters*, vol. 28, no. 2, pp. 214 – 223, 2007.

[103] WELLS III, W., VIOLA, P., ATSUMI, H., NAKAJIMA, S., and KIKINIS, R., "Multimodal volume registration by maximization of mutual information," *Medical Image Analysis*, vol. 1, no. 1, pp. 35 – 51, 1996.

[104] WESTBROOK, C., ed., *MRI at a Glance*. Malden, MA: Wiley Blackwell, 2010.

[105] WHITAKER, R. T., "A level-set approach to 3d reconstruction from range data," *International Journal of Computer Vision (IJCV)*, vol. 29, no. 3, pp. 203 – 231, 1998.

[106] YUSHKEVICH, P., PIVEN, J., HAZLETT, H., SMITH, R., HO, S., GEE, J., and GERIG, G., "User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage*, vol. 31, pp. 1116 – 1128, 2006.

[107] ZHOU, E., FU, M., and MARCUS, S., "A particle filtering framework for randomized optimization algorithms," in *Winter Simulation Conference, Proceedings of the*, WSC '08, pp. 647 – 654, Winter Simulation Conference, 2008.

[108] ZITOVA, B. and FLUSSER, J., "Image registration methods, a survey," *Image and Vision Computing*, vol. 21, pp. 977 – 1000, 2003.